Semantic Constraint Validation in Knowledge Representation for the Semantic Web: A survey, taxonomy and research challenges

Mariano Julián Ferreirone[a,b,c,*], Mario Lezoche[a], Chiara Franciosi[a], Hervé Panetto[a] and Diego Torres[b,c]

[a] CNRS, CRAN, University of Lorraine, Nancy, 54000, Grand Est, France
[b] LIFIA, CICPBA, UNLP, Facultad de Informática, B1900, La Plata, Buenos Aires, Argentina
[c] CyT, UNQ, Quilmes, B1818, Buenos Aires, Argentina

ARTICLE INFO

Keywords:
Knowledge Representation
Semantic Constraints
RDF Validation
Ontology
Knowledge Graph
Shapes Graphs

ABSTRACT

Semantic web models such as ontologies and knowledge graphs were designed to represent knowledge in an explicit way and to infer implicit knowledge. This brings several benefits like the flexibility and reasoning power. On the other hand, since the adoption of these models has been growing during the last years, a demand of consistency raised up in parallel. As a consequence, the validation of these representations started to be an important issue to research. Several types of related works have been done during the last years, however, for the best of the author's knowledge, there is still a lack of an integrated review of these works in order to understand how is conformed the state of the art of formal semantic constraint validation research. The contribution of this article includes a taxonomy which describes a set of categories of related research trends, the classification of the selected articles in these categories and the enumeration of the current open challenges on the validation of semantic constraints in knowledge representation for the Semantic Web. Each of these contributions was obtained through the conduction of a systematic literature review. In addition, an extension of the study is done in order to take into account a very promising model which is having a growing intersection with the field: property graphs. Finally, this paper concludes with an outlook that summarizes the contributions done and the challenges that the authors decided to continue researching in further steps.

1. Introduction

Berners-Lee has defined the semantic web (SW) as a decentralized extension of the web of documents, where the information would have a meaning and would be machine readable ([17]). A fundamental part of this web consists in the representation of the human knowledge through the creation of formalized semantic models, which are able to communicate each other, creating in this way an open, accessible, universal and connected bigger model. For this purpose, knowledge representation (KR) comes into the scene as a science that studies the representation of real-world knowledge from a specific domain ([48]). Furthermore, the formalization of these models is possible by the use of logic, such as Description Logic (DL). SW models represent knowledge in an explicit way, but also present the power to infer implicit knowledge. Two well-known models of these technologies are ontologies and Knowledge Graphs (KGs). An ontology represents the knowledge mostly at a conceptual level, and the most used language is the Web Ontology Language (OWL). On the other hand, a KG is based on the instance level, and Resource Description Framework (RDF) is the most widely used language to create them ([51]). Both models offer attractive features for the community, enabling flexible domain knowledge representation, global resource identification, implicit inference, and connection to previously published web models. As part of a previous work, the authors of the current article have made a comparison of both models' expressiveness in [57].

The main KR models are incomplete by nature and consequently, they represent knowledge following the Open World Assumption (OWA) and Non-Unique Name Assumption (NUNA). The former means that if there is some fact that is not represented in the model, it could not be taken as false, meanwhile the latter states that it is possible to represent a real world entity multiple times, with different names ([161], [15]). KR models are designed mainly for the

---

[*] Corresponding author

mariano-julian.ferreirone@univ-lorraine.fr (M.J. Ferreirone); mario.lezoche@univ-lorraine.fr (M. Lezoche); chiara.franciosi@univ-lorraine.fr (C. Franciosi); Herve.Panetto@univ-lorraine.fr (H. Panetto); dtorres@lifia.info.unlp.edu.ar (D. Torres)
ORCID(s): 0009-0008-5610-5086 (M.J. Ferreirone)

inference of new knowledge. When it comes to the consistency of the model and its validation, this means a problem. There are some domains, as medicine, that can't take the risk of creating their systems based on models that could not be consistent enough. This issue has opened a whole path of research that is trying to keep the benefits of the models and, at the same time, to add the consistency needed to increase the adoption of these technologies. Many works have been done with the intention to set these models with Closed World Assumption (CWA) and Unique Name Assumption (UNA). The objective of these works is to be able to validate the aforementioned models in a closed manner, similar to how it is done in Relational Databases (RDBs). In order to solve this problem, the focus is concentrated over the semantic constraints (SCs) that restrict the models, how they are represented and how they are interpreted. The study of this issue is key for increasing the adoption of not only the mentioned models, but also the whole SW. In consequence, it is interesting for the community to review what has been done in relation to this problem, which are the open paths that will guide the evolution of the developments and which are the related issues that benefit from these advancements.

The contribution of this article includes a taxonomy which describes a set of categories of related research trends, the classification of the selected articles in these categories and the enumeration of the current open challenges on the validation of semantic constraints in knowledge representation for the SW. Furthermore, these contributions include as well the study of Property Graphs, which present an interesting intersection with the SW and some of the problematics found in this study. Each of these contributions is obtained through the conduction of a systematic literature review which is guided by a set of research questions (RQs) that plan to investigate details such as the knowledge representation models (KRMs) involved in the works, the type of SCs that are being treated, the languages in which the SCs are being represented and the type of contributions developed in the selected articles. In addition, a discussion based on the results is presented, where the research trends which were more active during the last years and present more potential are highlighted. Finally, some conclusions are stated and the most interesting challenges found are defined as part of the future steps of the authors.

This paper is organized as follows: Section 2 introduces the background, where the fundamental concepts of this work are described. Section 3 presents the related works and Section 4 describes the systematic method that is applied for conducting the review. Section 5 presents the definition of a taxonomy and shows the obtained results. Section 6 represents a discussion over the integrated outcome of the review and describes the identified current open challenges. Finally, section 7 presents the conclusion of the paper and the definition of further steps.

2. Background

This section introduces the core concepts underpinning semantic constraints (SCs) and knowledge representation (KR), followed by their most common representation models, associated languages, and illustrative examples.

2.1. Core Concepts in Knowledge Representation

2.1.1. Knowledge Representation

KR has been defined in [39] as a surrogate, a set of ontological commitments, a fragmentary theory of intelligent reasoning, a medium for efficient computation, and a medium of human expression. This kind of representation contains a symbolic description of a part of the real world, in the form of statements that reflect the facts related to a domain. Furthermore, the use of sorts is a natural way to formally express certain knowledge types and roles at the knowledge level, as already advanced in [10]. These representations aim to simulate intelligent human behavior, which means that the knowledge represented could then be analyzed and processed by a machine through reasoning algorithms.

In the early stages of the Semantic Web (SW), KR appeared in different forms, the most prevalent of which was based on semantic networks, rules, and logic. The former could be found in RDF graph representations or Topic Maps, and they focused on expressing the taxonomic structure of categories of objects and the relations between them ([130]). Furthermore, these networks are closely related to another form of KR called frame systems, but these use different representation metaphors. With respect to the rules and the logic, they often aimed to represent the business knowledge in the form of business rules or logic programming formalisms ([158]). On the other hand, graphs have long been considered for knowledge modeling and for computation. A benefit of this approach is that labeled graphs, schemata and drawings provide an intuitive vehicle for KR ([125]).

More recently, the field has evolved towards Knowledge Graphs (KGs) as the dominant paradigm for representing and integrating heterogeneous data on the Web ([51], [84], [87]). KGs combine the expressiveness of ontologies (which provide the conceptual schema and formal semantics) with large-scale instance data structured as entities and semantic relations. They are typically implemented using RDF and related Semantic Web standards, which allow for flexible integration, reasoning, and querying at scale. This shift reflects the growing need to handle large, dynamic, and semantically rich data, which earlier KR models like Topic Maps and simple taxonomic structures could not fully address. As such, ontology and KG have become two of the most widely adopted graph-based representation models in current SW research and applications.

2.1.2. Ontology

A well-known KR model is the ontology, which provides the means for describing explicitly the concepts behind the knowledge represented in a knowledge base (KB, [67]). An ontology is a formal, explicit specification of a shared conceptualization, describing the vocabulary used to model a domain ([70]). There, the knowledge is formalized using five kinds of components: classes, relations, functions, axioms, and instances ([69]). Furthermore, rigorously formal ontologies are defined in a language with formal semantics, theorems, and proofs of such properties as soundness and completeness ([168]). There are three different levels of knowledge abstraction that need to be formalized: methodological knowledge (ontology language), conceptual level (classes and their relationships), and factual knowledge (individuals and their relationships). Finally, ontologies are mostly used to represent the conceptual level of the knowledge (TBox), creating a kind of schema that represents the bases of a specific domain (there are many examples as [35], [14] and [49]).

In order to exemplify a very simple ontology, let's think of a university. An ontology for this domain might specify that:

- There are classes such as Student, Professor, and Course.
- There are relationships such as teaches (from Professor to Course) and enrollsIn (from Student to Course).
- There may be constraints, such as "Every course must be taught by exactly one professor."

This ontology does not contain actual people or specific courses but only the conceptual structure describing how such entities relate to each other.

2.1.3. Knowledge Base

A Knowledge Base (KB) is a structured repository that integrates conceptual knowledge (the TBox) with factual knowledge about specific instances (the ABox). The conceptual knowledge is typically expressed using an ontology or schema that defines the classes, properties, and constraints of a domain. The factual knowledge consists of assertions about individual entities and their relationships according to this schema [75].

In addition to schema and instance data, KBs often include rules, constraints, or logical constructs that enable automated reasoning and inference, allowing implicit knowledge to be derived from explicitly stored facts. Thus, a KB serves as both a data store and a basis for reasoning, enabling complex queries and knowledge retrieval.

In order to exemplify, let's continue with the university domain. The ontology (TBox) might define classes such as Student, Course, and Professor, and properties like enrolledIn or teaches. The instance data (ABox) could include assertions like Alice is a Student, Bob is a Professor, and Alice enrolledIn Course101. Together, these facts and the schema form a knowledge base that can answer queries such as "Which professors teach courses that Alice is enrolled in?"

2.1.4. Knowledge Graph

A KG is a formal representation of knowledge in the form of a labeled directed graph, where the nodes represent concepts or actual entities from the real world, while the edges represent different relations between these nodes ([84]).

Historically, the notion of KGs emerged alongside the evolution of the SW. Initially, the SW was driven by the vision of a Web of Data, realized through the Linked Data principles proposed by Berners-Lee, aiming for publishing and interlinking data ([18]). In this context, some prominent KGs (such as DBpedia and Wikidata) were designed as part of the Linked Open Data cloud, enabling large-scale integration and semantic querying.

However, as the field progressed, the notion of KGs broadened beyond strictly Linked Data. Many modern KGs (particularly enterprise Knowledge Graphs) have been developed internally to integrate diverse data sources and support specific business applications. These KGs are typically not published as open-linked data. As discussed by Hitzler (2021), this reflects the historical development of the SW field, which can be roughly divided into three phases: the original vision phase, the standardization and ontology engineering phase, and the adoption phase marked by the growth of Linked Data and the widespread emergence of Knowledge Graphs, both open and private ([83]). Therefore, while some KGs are part of the Linked Data landscape, it is more accurate to describe KGs as a broader class of knowledge representation artifacts that may or may not conform to Linked Data principles, depending on their scope and intended use.

Additionally, KGs may include an underlying schema to increase expressiveness, which is the case when they are ontology-based. KGs built on shared ontologies are more interoperable, since the ontology structure is unambiguous and has an accepted, common meaning within the community ([84]).

In the university domain, a Knowledge Graph would represent entities such as Alice (a student), Course101 (a course), and Bob (a professor) as nodes. The relationships between them (such as Alice enrolledIn Course101 and Bob teaches Course101) would be edges connecting these nodes. Each node and edge can have rich metadata (labels, types, properties).

2.1.5. Semantic Constraints

The SCs are a fundamental part of the KR model. They can be seen as rules applied to the models which define logical terms with the objective of defining logical consequences ([138]). Ontology modeling languages allow defining these constraints on classes, properties, or on the entire ontology, and this increases the expressiveness that the model offers ([8]). These constraints could be of different types, with different objectives, such as the cardinality constraints, which restrict the amount of relationships that a set could have with another set (for example, an exam could have only one score). Furthermore, the object property path constraints present an example of a more complex restriction which states that if an individual x is connected through a chain of object property relationships $r1, r2,..,rn$ with another individual n, then the individual x is directly connected to the individual n with this type of property ([26]). In general, the RDF model respects the OWA and NUNA, paving the way to the inference of new implicit knowledge as the result of logical consequences based on the defined SCs.

2.2. Representation Models

2.2.1. Resource Description Framework (RDF)

RDF is the standard data model for representing knowledge in the web of data and in the Linked Open Data[1]. This framework is defined as graph-based and its syntax follows the structure of a set of triples of the form (subject, predicate, object). A set of these triples is called an RDF graph, where the subject and the object are represented as nodes; meanwhile, the predicate is represented as a directed relationship which goes from the subject to the object. Furthermore, the nodes could be represented as IRIs, literals and blank nodes (bnodes). The first two denotes something in the world, having the IRIs as a unique way of representing a resource, and a literal that refers to a specific data type which can only be represented by object nodes. On the other hand, the bnodes represent the existence of something that is not identified. RDF supports simple semantics, and it's based on XML, so it inherits the XML datatype definitions. A graphical representation of a simple RDF graph is shown in picture 1.

2.2.2. RDF Schema (RDFS)

The Resource Description Framework Schema (RDFS) is a semantic extension of RDF, which provides a vocabulary for increasing the expressiveness of these models. This framework allows defining classes for describing groups of related resources or entities. Moreover, this hierarchy could also be defined for the relationships that are represented as predicates. Finally, the types of subjects and objects related to a specific predicate can be determined through the definition of domain and range properties, respectively. The specification of this framework can be found in [28].

2.3. Languages and Standards

2.3.1. SPARQL Protocol and RDF Query Language

SPARQL[2] is the standard query language defined for querying RDF data ([36]). This language is useful for consulting and modifying the data which is stored in one or more RDF graphs, where the data is stored in a triple RDF format. In other words, this language contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. In addition, there are different kinds of queries that could be created and executed, such as SELECT which directly returns variables and their bindings, CONSTRUCT which returns a single RDF graph, ASK that is useful only for returning a boolean answer to whether the data consulted exists or not

---

[1] LOD, https://www.w3.org/egov/wiki/Linked_Open_Data

[2] https://www.w3.org/TR/rdf-sparql-query/ (last accessed on 2/11/2025)

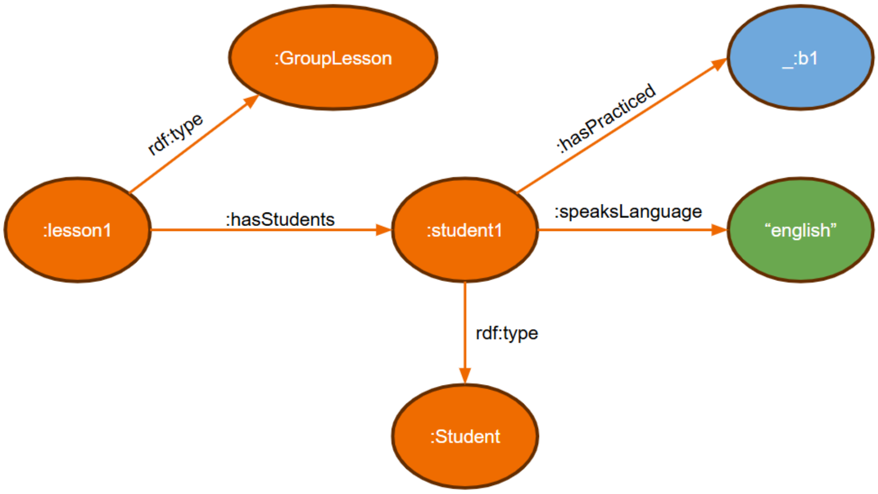

Figure 1: A graphical representation of a simple RDF graph example. The nodes in orange represent IRIs, the green one represents a literal and the light blue a blank node. This graph represents knowledge related to ski lessons of type groupLesson. In addition, the graph presents knowledge related to a student who participated in an instance of the aforementioned lesson.

and finally DESCRIBE returns a single result RDF graph containing RDF data about resources. The definition of ASK SPARQL queries has been used as an alternative approach for constraining a knowledge representation model in a closed manner, following Closed World Assumption (CWA).

2.3.2. Web Ontology Language

Ontologies can be expressed in different description languages, and each of them means a trade-off between expressiveness and complexity of reasoning ([105]). In the Semantic Web community, the ontologies are commonly represented with the Web Ontology Language (OWL)[3] and its family, which is a knowledge representation language based on description logic ([100]). These languages present an interesting level of expressiveness, quite wider than the one offered by RDFS, allowing to represent characteristics of the resources and the properties such as the cardinalities, the type of properties depending on their behavior (transitive, symmetric, reflexive, etc.) or on their type of ranges (object property or data property), and much more. OWL has a rich vocabulary with several features, but in this section, only a few relevant properties are briefly described.

To start, a class in OWL (*owl:class*) is an abstract representation of a group of individuals with similar characteristics. In addition, the classes are defined with *class axioms* which could contain different constructs for the description of the characteristics and its hierarchical relationship with other classes. One of the constructs useful for the creation of a class axiom is *owl:equivalentClass*, which allows to define that the class being defined is equivalent to another class. Furthermore, the class axioms could include *class descriptions* of different types, such as *property restrictions*. A property restriction describes a class as the set of individuals that satisfy the proper restriction. An example of a property restriction is the *owl:maxCardinality* which limits the amount of allowed individuals of the class. Finally, OWL allows to define *properties* which could be of two different types: *Object Properties* (which link individuals to individuals) and *Datatype Properties* (which link individuals to data values). While RDF, RDFS, and OWL provide modeling capabilities, they do not enforce closed-world validation. Constraint Languages are useful for addressing this gap.

2.3.3. Constraint Languages (CLs)

Many SW models are graph-based and lack a rigid semantic schema, making them unsuitable for validation. This lack of schema, along with OWA and NUNA, hinders closed validation. To address this, defining CLs as add-ons to existing models has garnered significant attention as a solution ([166]). These new languages follow the design of a set of directed constraints which aim to validate the data following CWA and Unique Name Assumption (UNA)

---

[3] https://www.w3.org/TR/owl-guide/ OWL (last accessed on 2/11/2025)

configurations. The most popular CLs are the Shapes CL, having in SHACL ([134]) the current standard defined by the World Wide Web Consortium (W3C)[4], and in the Shapes Expressions (ShEx) ([149]) language another popular example. These "Shape Graphs" allow verifying the consistency of the graph instance data against a designed shape, which represents the desired structure of the graph.

An example of a SHACL shape is shown in the listing 1. The first two lines define the prefixes that are used in the shape, where *sh* is the prefix for the SHACL standard language and the prefix *ex* aims to an ontology from which the properties and classes are referenced. This shape is identified with the IRI ex:InstructorShape. The line 5 defines the target of the shape as those instances of class ex:instructor. Next, line 7 states that the shape will constraint the value nodes vn which are connected through the property ex:teaches_discipline. Besides, line 8 defines the sh:class value type constraint, which restricts the value nodes to those instances of class ex:Discipline. Finally, line 9 defines a minimum cardinality of one. In summary, this shape can be interpreted as a profile of instructors who must teach at least one discipline.

Listing 1: Example SHACL shape

```
@prefix sh:    <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .
ex:InstructorShape
    a sh:NodeShape ;
    sh:targetClass ex:Instructor ;
    sh:property [
        sh:path ex:teaches_discipline ;
        sh:class ex:Discipline ;
        sh:minCount 1 ; ] .
```

This is a topic that has been attracting the industry and the academics during the last years ([96]). Several applications of these languages have been proposed, such as the validation of a SW based system ([42], [169]) or the controlled evolution of KRMs ([29]).

2.4. Comparison of Ontology, Knowledge Base, Knowledge Graph, and RDF Dataset

The terms ontology, knowledge base (KB), knowledge graph (KG), and RDF dataset are closely related within semantic technologies, yet they serve distinct roles and have different scopes. In this subsection, there is a brief summarization of these concepts and a highlight of their differences and relationships.

- Ontology is a formal, explicit specification of a shared conceptualization, describing the vocabulary (classes, relations, axioms) used to model a domain ([70]). Ontologies primarily define the TBox, which forms the schema layer enabling reasoning and semantic interoperability.

- Knowledge Base (KB) is a broader concept referring to a structured repository that combines the TBox (schema, usually provided by an ontology) and the ABox (instance data or factual assertions). As van Harmelen et al. (2008) describe in [75], a KB may also include rules and other logical constructs to support inference and querying.

- Knowledge Graph (KG) is a graph-structured representation of real-world entities and their semantic relationships, often constructed by instantiating an ontology with concrete data ([51]). A KG typically includes both schema and instance data and emphasizes graph traversal and semantic queries.

- RDF dataset is a technical serialization of semantic data according to the Resource Description Framework (RDF) standard ([36]). RDF datasets can encode instance data or complete knowledge graphs, and serve as the concrete data format for storage, exchange, and integration.

Although these concepts overlap (for example, a KG can be viewed as the ABox populated over an ontology, and stored as an RDF dataset), they each play a distinct role in semantic systems: ontologies define the conceptual model; knowledge bases aggregate schema and data (and possibly rules); knowledge graphs structure that data as entities and relations for graph-based exploration; and RDF datasets provide the standard format to serialize and share all of the above.

---

[4] W3C, https://www.w3.org/

3. Related work

There are three works that could be taken as related works to this article, and that are briefly described in this section. The first article found was "Requirements on RDF Constraint Formulation and Validation" ([78]), which presents a review of the existing requirements related to the definition and validation of SCs. It presents an analysis of many types of constraints and CLs and defines if they can manage the aforementioned requirements. The second work identified was "RDF Validation: A Brief Survey" ([166]), which shows a description and comparison of the state of the art CLs at the moment. Finally, the third related paper is "Challenges in RDF validation" ([96]), which presents several challenges that are interesting to research related to RDF validation. However, most of the article is dedicated to the comparison of two existing CLs, ShEx and SHACL.

Even if the aforementioned articles are interesting and present valuable information related to the validation of SCs on KRMs for the SW, none of them are based on a systematic literature review (SLR) method. Conducting an SLR brings well-known advantages, such as transparency, reproducibility, and reduction of bias through predefined research questions and selection criteria. Beyond these general benefits, an SLR is particularly valuable for the tackled issue because the field of validating constraints on KRMs is highly heterogeneous: it covers a wide range of knowledge representation models (including ontologies and knowledge graphs), multiple constraint languages with different expressiveness and semantics, and approaches that often intersect with other research areas like reasoning and interoperability. This diversity, combined with the rapid emergence of new methods and languages (for example SHACL, ShEx, and their extensions), makes it challenging to capture the full state of the art with a traditional narrative review. Therefore, by applying a systematic approach, this work aims to provide a comprehensive, up-to-date, and balanced synthesis of the field, identifying gaps and trends that can inform future research.

Going forward with the differences, the approach of [78] is limited to the study of the constraint formulation and expressive capabilities of existing CLs. The core of the work presented in [166] is quite similar, since the objective is to define the requirements that a CL should fulfill, and the overview is limited to existing approaches defined in RDF validation. Moreover, in [96] the review is restricted to a comparison between the core versions of ShEx and SHACL languages through a rewriting of both to a new intermediate language. These three related works differ from the current article, which (to the best of the author's knowledge) is the first SLR focused on the validation of SCs on KRMs for the SW from a broad point of view, covering all models, constraints, constraint languages, and also taking into account the intersection between solutions for this issue and other well-known challenges. These issues include: recursion, which concerns the ability of constraint languages to express self-referential structures; satisfiability, which examines whether a given set of constraints can be jointly satisfied on any data instance; containment, which studies the relationships between different constraint sets (whether one is more restrictive than another); and expressiveness, which assesses the range and complexity of constraints that can be formulated. Further, the review considers mappings and interoperability, which explore how constraints facilitate data integration across heterogeneous schemas; schema extraction, which looks at deriving constraints automatically from existing data; and validation reports and explanations, which analyze how tools communicate violations and guide users in correcting data. Other key issues include error treatment (how tools and languages handle inconsistencies or partial validations), reasoning (how logical inference can support or complicate validation), query optimization (how constraint knowledge can help improve query performance), and the design of validation algorithms themselves.

Investigating these intersections is essential not only for understanding the technical limitations and design trade-offs in constraint validation but also for identifying the broader usages of semantic constraints beyond mere validation (for instance, improving data integration, supporting reasoning tasks, enabling data cleaning, or driving schema evolution). By systematically analyzing how validation techniques and semantic constraints relate to and address these issues, the review offers a deeper and more comprehensive picture of the role that semantic constraints play in the evolving Semantic Web ecosystem.

Given that the validation of semantic constraints is closely related to data quality assessment in the Semantic Web, it is important to clarify their respective scopes, the areas where they overlap, and how they differ. Quality assessment is typically broader in scope: it aims to evaluate multiple dimensions of data quality, such as completeness, accuracy, consistency, timeliness, and conciseness, often using a mix of heuristics, statistical analysis, and domain-specific metrics ([174]). In contrast, constraint checking focuses specifically on verifying whether data instances comply with explicitly defined semantic constraints, such as shape expressions, cardinality restrictions, or logical axioms formulated in constraint languages or ontologies ([95]; [128]).

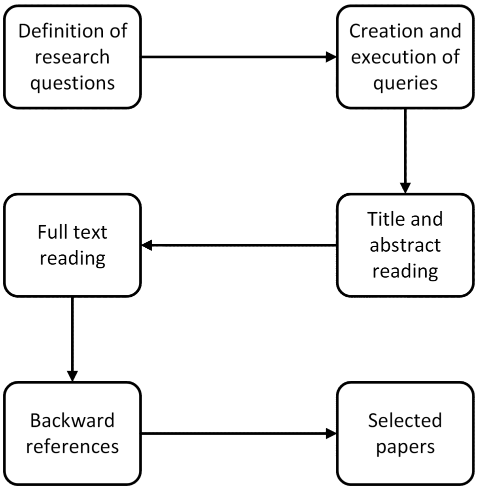

Figure 2: The SLR Steps

The overlap between these areas lies in the fact that constraint checking can serve as one method to detect quality problems: violations of semantic constraints can reveal errors, inconsistencies, or omissions in the data ([92]). However, quality assessment usually goes beyond constraint checking, incorporating analyses that do not necessarily rely on explicit constraints; for example, detecting outliers, measuring coverage, or assessing adherence to best practices ([174]). By clarifying this relationship, the present review focuses specifically on constraint checking as a formal and declarative approach, while recognizing its role as a fundamental component within the broader context of data quality management for the Semantic Web.

4. Method

This work presents an SLR methodology inspired in the work presented by Petersen in [126], with the objective of reviewing the scientific work in a research area and identify the state of the art. Similar reviews have been applied in different published works such as [37], [61] and [141].

The main aim of this review is to identify and analyze all the relevant works associated with the **validation of SCs related to KRMs for the SW** and the extension of the study to include Property Graphs. Additionally, articles may not only focus on validation but also on combining it with other techniques like Query Optimization (QO). The review also analyzes research gaps and outlines open challenges in the domain, providing a clear understanding of validation challenges in SW's KRMs and suggesting future directions for development. The figure 2 describes the steps of the SLR method, which are detailed in the next sections.

4.1. Research Question

The goal of this review is to identify all the relevant literature related to the validation of SCs in KRMs for the SW. This leads to the following definitions:

The main RQ:

- **RQ:** Which are the existing trends related to the validation of SCs in KRMs for the SW?

Specific RQs:

- **RQ1:** Which are the KRMs involved?
- **RQ2:** Which are the types of contributions presented? This includes, for example: Definition of CLs, mapping techniques.
- **RQ3:** Which types of SCs are represented?
- **RQ4:** Which languages are used for the representation of the SCs?

Table 1
Groups of keywords

Group *A*	Group *B*
Semantic constraint	Knowledge representation
Semantic restriction	Knowledge formalization
Ontological constraint	Semantic network
Shapes	RDF
SHACL	Ontology
SHEX	Knowledge graph
	OWL

4.2. Keywords and search string definition

The keywords definition is done with the intention of giving a structure to the search, in line with the main RQ defined. The identified main keywords are *"semantic constraints"* and *"knowledge representation"*. In consequence, the creation of the query must consist of two groups (*A* and *B*) connected with the Boolean operator AND. The groups are detailed in table 1 and a general final query was defined as:

("Semantic constraint" OR "Semantic restriction" OR "Ontological constraint" OR Shapes OR SHACL OR SHEX) AND ("Knowledge Representation" OR "Knowledge formalization" OR "Semantic Network" OR RDF OR OWL OR Ontology OR "Knowledge Graph")

The selection of the keywords is done after testing several combinations of potential keywords, analyzing the influence that they had in the obtained results. *"Semantic constraints"* and *"Semantic restrictions"* are taken as synonyms, since most of the literature uses them interchangeably. *"Ontological constraints"* represents constraints which are represented in an ontology. The *"Shapes"* keyword is related to a popular SC format, which is represented by CLs. The keywords *"SHACL"* and *"ShEx"* represent the two most popular languages used for represent shapes constraints. Terms related to *"Shapes"* are added to the query since the authors were aware of the popularity of these techniques for the creation of SCs at the time when this review was started. Regarding the group *B*, the term *"knowledge representation"* refers to the artificial intelligence domain in which the models of interest are studied, meanwhile, *"Knowledge formalization"* aims to those models which respect a semantically structured framework. The keywords *"Semantic network"*, *"Ontology"* and *"Knowledge graph"* target the respective specific representations. Finally, the keywords *"RDF"* and *"OWL"* are added, since they represent very well-known languages for the representation of the aforementioned models.

4.3. Database selection

Following the advice given in [91], four scientific databases were selected for this work: Scopus, Web of Science (WoS), IEEE Xplore and ACM. These digital libraries cover different scientific fields, including peer-reviewed articles related to computer science, SW journals and conferences. Furthermore, they offer advanced search features that allow to create customized queries, what is valuable for creating a reproducible set of papers.

The reference management tool Zotero[5] was used to store and organize the obtained papers. In addition, the collaborative online tool Scolr [6] was used to delete the duplicates, discard those articles that have incomplete information (title, abstract or year) and classify the papers. This study was conducted on September 2023, so the related articles published up to that moment are taken into account.

4.4. Study Selection and Quality Assessment

There are four filtering stages that are applied, based on the ICRs and ECRs defined in table 2. There, the term "Entire conference proceedings volumes" (in ECR1) refers specifically to excluding the proceedings volumes themselves (i.e., collections that compile all the papers accepted at a conference or workshop) from being treated as primary studies in the review. Instead, the review only considers individual full-length papers published within these

---

[5] Zotero, https://www.zotero.org/

[6] Scolr, https://scolr.lifia.ar/

Table 2
List of inclusion and exclusion criteria, having ICRn = 'Inclusion Criteria n' and ECRm = 'Exclusion Criteria m'.

Criteria	Description
ICR1	The article is related to the validation of SCs on KRMs
ICR2	The article is related to a formal and semantic CL
ICR3	The articles are in the field of computer science
ECR1	Entire conference proceedings volumes and entire books (individual peer-reviewed papers from these may be included); short papers, demo papers, and posters
ECR2	Studies presenting non-peer reviewed material
ECR3	Studies not presented in the English language
ECR4	Studies with incomplete information (title, abstract, keywords)
ECR5	Studies not accessible in full-text
ECR6	Studies that are duplicates of other studies
ECR7	Papers that were published before the year 2001

Table 3
Amount of obtained papers by scientific database

Database	Total	Unique	Duplicated
Scopus	1054	1054	0
IEEE Xplore	524	375	149
WoS	606	191	415
ACM	531	177	354
Total	2715	1797	918

proceedings, provided they meet the other inclusion criteria. The rationale for this choice is that proceedings volumes themselves are secondary compilations and do not present original research contributions in the form of single studies, and thus fall outside the scope of primary research required for the SLR. Furthermore, it is worth to mention that the ECR7 is defined taking into account that the first OWL specification was done in 2004 ([170]), so it is not expected to find articles related to the validation of KRMs before that year. In order to have some extra margin, the search is done starting from 2001. The four stages are: filters on databases (creating an adapted final query per database), discarding duplicates through Scolr, reading the title-abstract and reading the full-text. In addition, more articles are added through the application of backward snowball sampling.

The execution of the queries with the filters retrieved an initial set of 2715 papers which were reduced to 1797 after removing the duplicates. The detail of the obtained papers per database and their uniqueness is detailed in table 3. In this table, the *'Duplicated'* column reflects the amount of duplicated papers with respect to the set of papers that were previously loaded into Scolr. As an example, the first row describes that no duplicates were found between the papers retrieved from Scopus, meanwhile the second row presents that 149 papers from IEEE Xplore were already retrieved on the Scopus set. Going forward, after the title-abstract reading, 1634 articles were excluded and 163 were selected for the following stage. These papers were reduced to 92 articles as a result of the full-text reading filter. Finally, the obtained papers were used to conduct a backward snowball sampling ([86]). This represented a full text reading of the references that were present in the papers aforementioned. Through this stage, 31 articles were added, creating a final selection of 123 papers. The detail of the amount of papers included/excluded on each stage is detailed in the figure 3. The selected papers are detailed by id in tables 4, 5 and 6.

Table 4
SLR selected papers 1

Article ID	Article	Article ID	Article
1	[24]	30	[47]
2	[104]	31	[147]
3	[171]	32	[172]
4	[71]	33	[173]
5	[68]	34	[115]
6	[73]	35	[119]
7	[152]	36	[162]
8	[2]	37	[148]
9	[165]	38	[143]
10	[34]	39	[133]
11	[30]	40	[155]
12	[59]	41	[94]
13	[19]	42	[156]
14	[33]	43	[101]
15	[31]	44	[154]
16	[96]	45	[21]
17	[23]	46	[85]
18	[98]	47	[144]
19	[45]	48	[7]
20	[79]	49	[40]
21	[43]	50	[111]
22	[114]	51	[72]
23	[50]	52	[5]
24	[128]	53	[53]
25	[120]	54	[122]
26	[64]	55	[52]
27	[55]	56	[20]
28	[121]	57	[129]
29	[46]	58	[176]

4.5. Data Extraction

The table 7 details the items that were extracted from the selected papers. These extractions were intended to classify the content and order it aiming for a better understanding of the domain and helping for answering the RQs previously detailed.

5. Results of the review

In this section, the findings done as an outcome of the SLR are presented, and the defined RQs are answered. The papers obtained from the SLR were published from 2008 to 2023 which shows the continued activity that this topic has presented during the last sixteen years. Furthermore, it is interesting to highlight the growth in the published amount of articles starting at 2014 that is related to the publication of the popular CLs ShEx and SHACL, which were defined to validate the SW models ([96]). The papers by year are represented in the figure 4, where the *y* axis represent the year of publication and the *x* axis shows the amount of articles. In addition, the figure 5 shows the venue

Table 5
SLR selected papers 2

Article ID	Article	Article ID	Article
59	[3]	88	[27]
60	[22]	89	[106]
61	[135]	90	[167]
62	[118]	91	[66]
63	[41]	92	[78]
64	[65]	93	[25]
65	[136]	94	[62]
66	[153]	95	[151]
67	[58]	96	[161]
68	[102]	97	[13]
69	[164]	98	[123]
70	[74]	99	[9]
71	[108]	100	[124]
72	[82]	101	[11]
73	[97]	102	[92]
74	[157]	103	[99]
75	[142]	104	[146]
76	[103]	105	[150]
77	[54]	106	[38]
78	[160]	107	[60]
79	[159]	108	[89]
80	[44]	109	[112]
81	[132]	110	[166]
82	[63]	111	[109]
83	[56]	112	[113]
84	[163]	113	[127]
85	[139]	114	[77]
86	[175]	115	[88]
87	[6]	116	[1]

distribution of the papers, where the x axis describes the types of venues found and the y axis represents the amount of articles. Furthermore, 43 different conferences have a presence in the results, having the International Semantic Web Conference (ISWC) and the European Semantic Web Conference (ESWC) as the two most frequent.

5.1. Inclusion of property graph models in the results

During the systematic literature review, it was identified that although the Semantic Web (SW) stack is formally centered on standards such as RDF, some of the recent works address problems related to the semantic constraint validation in property graph (PG) models. While PGs are not officially part of the W3C SW standards, there are compelling reasons to include them in the results of this review (limited to the categories where relevant related studies were found) as supported by recent literature.

Firstly, recent works propose a shape language for PGs inspired by SHACL, offering formal semantics and complexity analysis akin to RDF validation ([147]). Secondly, there is a growing interoperability trend between RDF and PGs: Hartig ([76]) presents formal mappings and translations between RDF* and PGs, highlighting that hybrid

Table 6
SLR selected papers 3

Article ID	Article	Article ID	Article
117	[81]	121	[107]
118	[4]	122	[140]
119	[32]	123	[117]
120	[116]		

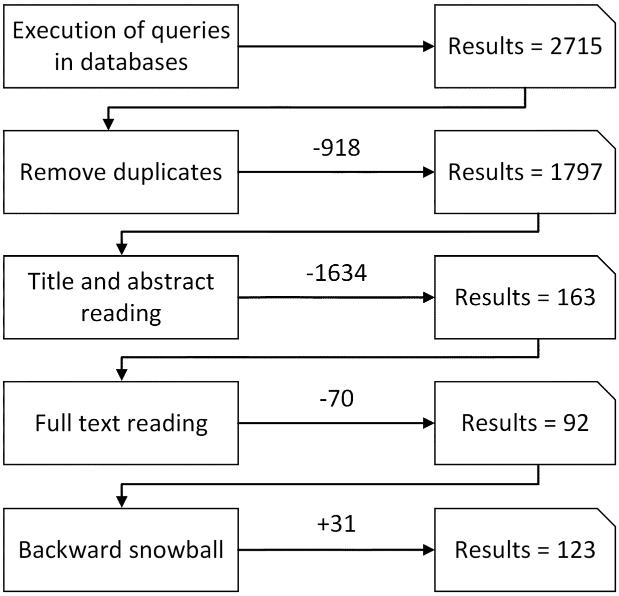

Figure 3: Number of articles included and excluded in each step of the review

Table 7
Data extraction description

Data Item	Value	RQ
Study ID	Integer	-
Title	Name of the article	-
Year	Calendar year of the publication	-
Venue	Conference, Journal, Workshop, Book chapter	-
Model	Ontology, KG,..	RQ1
Contribution	Type of contribution	RQ2
Constraints languages	SHACL, ShEx, SPIN, etc	RQ3
Constraints involved	Cardinality, Range, etc	RQ4

solutions increasingly combine the strong semantics of RDF with the flexibility of PGs. Angles et al. ([12]) further show how PG query languages have evolved under the influence of semantic data management needs, demonstrating conceptual convergence. Finally, recent surveys and industry analyses ([84]) confirm that enterprise knowledge graphs frequently rely on PGs for internal data integration and governance purposes, even if these graphs are not published as

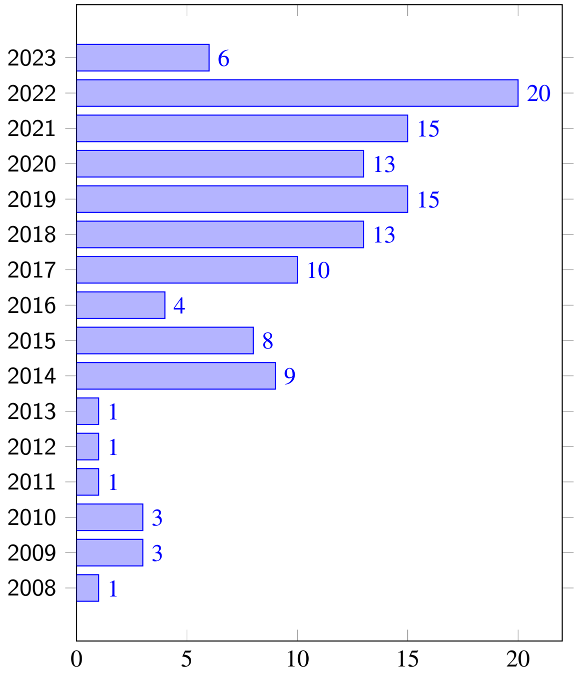

Figure 4: SLR selected papers by year

open linked data. This indicates that semantic constraint validation is practically relevant across graph-based KRMs, not only those strictly within the SW stack.

In summary, including these works in the results offers a more comprehensive view of current approaches to constraint validation. This choice reflects the field's evolution and aligns with best practices in systematic literature reviews ([90]), ensuring that the study encompasses both formal standards and practically adopted graph models.

5.2. KR models involved (RQ1)

This section presents the KR models that are being validated in the selected papers. The found SW models are the RDF graphs, ontologies, KGs, RDFS graphs and KBs; meanwhile, beyond the SW, property graphs (PGs) are taken into account. The former five are already described in the section 2. With respect to the PGs, this data model is used to represent and store information through a network formed by nodes, edges and key-value pairs which are related to both nodes and edges ([137]).

Almost the half of the models (64) was based on an RDF graph which represents RDF data or an RDF dataset, and were directly classified as an RDF graph. Not very far in the quantity (44), the second most used model is the ontology, which is followed by KGs (21) in the third place. Finally, KBs (8), RDF/s graphs (4) and PGs (2) had a minor presence in comparison with the already mentioned models. The detail of the KR models used is presented in the figure 6.

The high use of the RDF data model in the validation related works was expected, knowing that this is the standard model defined for the SW and that it represents mostly instance data, which creates an interesting scenario for validation research. Furthermore, the amount of papers that presented their work based on ontologies reflects on the one hand that the inference of these models doesn't ensure consistency, and on the other hand, that their expressiveness power is enough valuable for being on the priorities for further developments. Moreover, KGs are relatively new models, and their validation took on more importance from 2017. Since they represent mostly instance data, in order to validate them, it would be necessary to have an ontology behind, or to create a validation schema by the extraction of constraints from the graph. Going forward, the papers in which the input was a KB presented these repositories mostly based on the RDF model. Finally, the low amount of published works dedicated to the validation of PGs presents evidence of an interesting research challenge that is emerging.

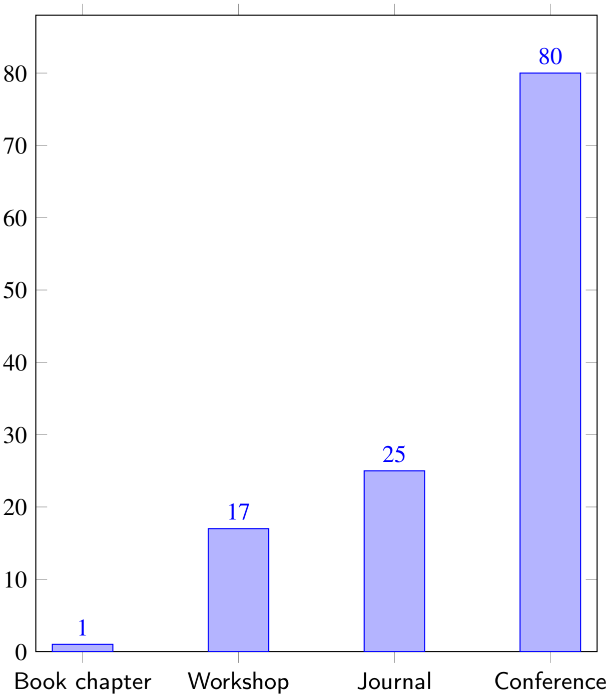

Figure 5: SLR selected papers by venue

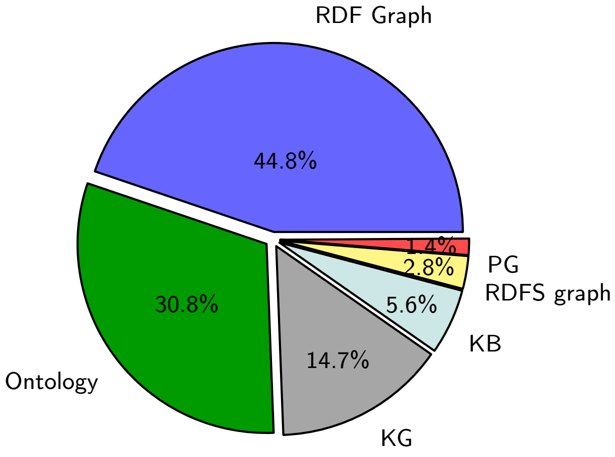

Figure 6: SLR selected papers by validated model

5.3. Type of contributions (RQ2)

In order to be able to answer this RQ, first, the definition of a taxonomy was done. This taxonomy represents the research trends found related to the validation of formal SCs and can be seen in the figure 7. The amount of papers by research trend is shown in figure 8. In this bar graph, the x axis represents the amount of papers, meanwhile the y axis represents the title of the taxonomy's categories. Each category is explained in more detail in the following subsections.

5.3.1. CL analysis and review

This category represents those works which propose an analysis or review of one or more existing CLs. The group of selected papers present six works that were classified into this category. There are some works which analyze a

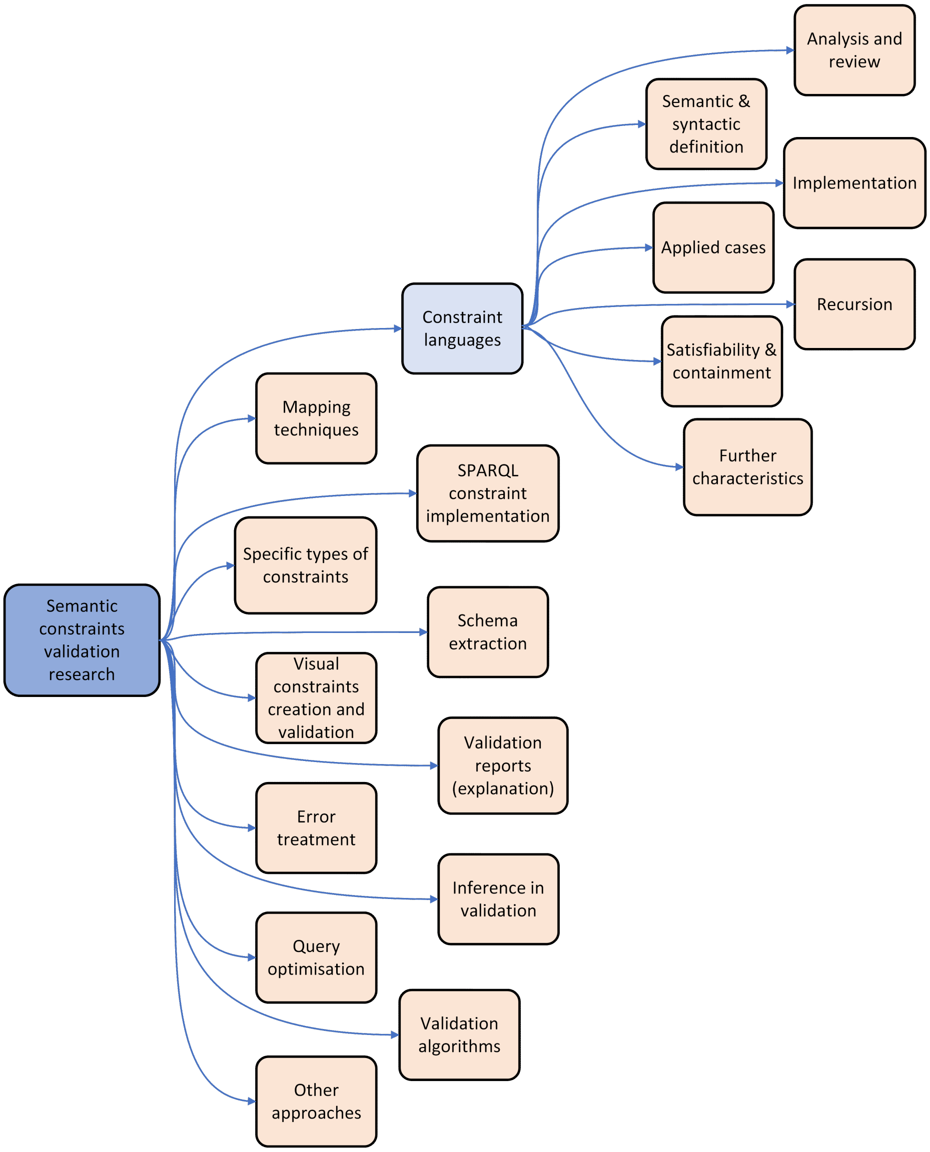

Figure 7: Taxonomy of research trends related to validation of formal SCs

single CL, focusing on the most important characteristics and features that it offers as in [164] and [119]. Furthermore, other type of analysis that can be found is through a comparison between two CLs, or between a CL and another type of language. This can be seen in [20], where SHACL is analyzed through a comparison with DLs, meanwhile in [96] SHACL and ShEx languages are analyzed by doing a comparison between both of them. Finally, there are works in which the analysis is based on the comparison between more than two languages, and their ability for the support of certain requirements related to RDF validation ([166], [78]).

5.3.2. *CL semantic and syntactic definition*

This type of work represents the definition of semantic and syntactic characteristics related to a CL. Fourteen of the selected papers were classified as part of this category. Some of these papers are based on the definition of a CL such as RDF Data Description language (RDD, [146], [59]), ShEx ([128]) and Wikidata Shapes Expression language (WShex, [66]). On the other hand, an important amount of these papers is focused on the definition of a semantic extension over an existing CL, which normally consists in the addition of new features to the respective language. The extensions over

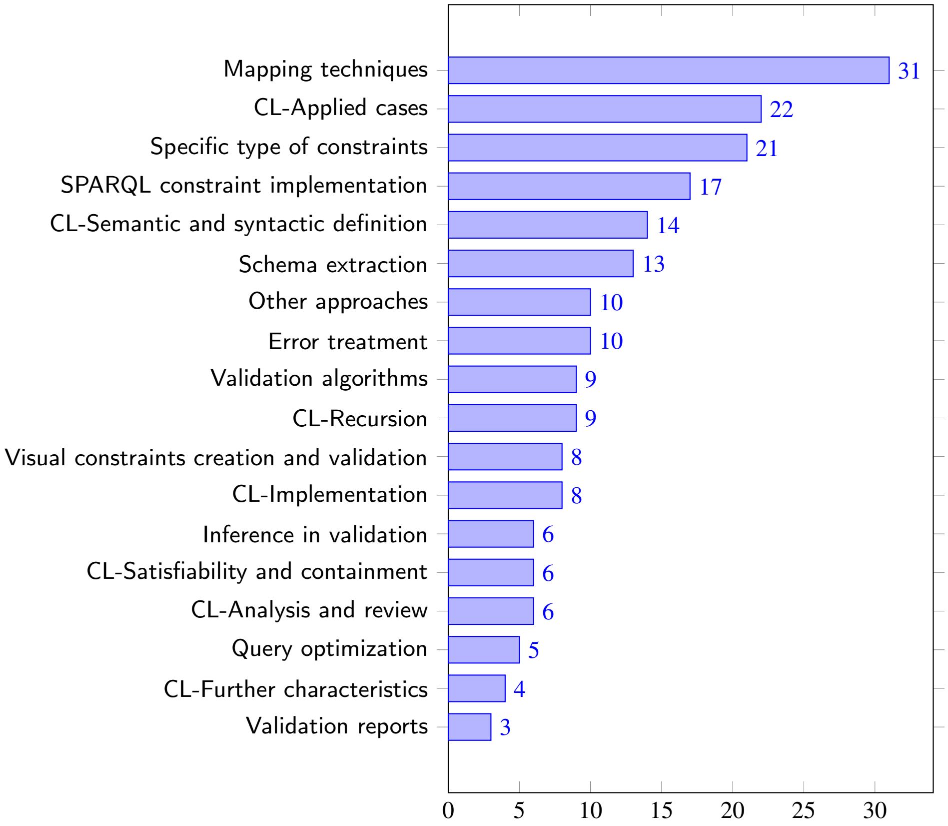

Figure 8: Distribution of the selected papers by their type of contribution (CL stands for 'Constraint Language')

ShEx are aiming to the definition of the 2.0. version and the support for single-type and multi-type features ([155], [22]). Meanwhile, the extensions to SHACL mostly aim to define new features such as recursion, and in some cases were inspired by other models such as logic programs, Answer Set Programming (ASP), and Approximation Fixpoint Theory (AFT) ([34], [11], [19], [30] and [21]). Another feature that started to capture the attention for the extensions in SHACL is the data provenance ([44]). Most of the aforementioned articles are explained in more detail in the categories which refer specifically to the CLs features related works. In addition, the totality of the already mentioned papers was related to RDF graph based CLs which were defined to constrain ontologies and KGs. Moreover, there is also a trend which aims to define the semantics and syntax of CLs which are defined to constrain PGs, which are more expressive KR models ([77] and [147]).

5.3.3. CL implementation

This type of work aims to the actual implementation of the semantics and syntax of a CL. Eight papers from the SLR outcome were classified into this category. An implementation of ShEx was found, having the inspiration in regular expression derivatives, meanwhile a review over four previously defined ShEx implementations was also presented ([65] and [128]). The implementation of SHACL is another objective of the community, which could aim to avoid the validation of invalid entities ([58]) and the computation of shape fragments ([44]). In addition, the definition of a benchmark that allows to test existing SHACL implementations in specific graph databases, is found quite necessary, and only one proposal was found up to date ([143]). Furthermore, SPARQL can serve as an effective execution layer for implementing RDF constraint languages (CLs), through the creation of queries that retrieves a closed answer, as demonstrated by existing approaches for Resource Description Rules (RDD, [59]) and Description Set Profiles (DSP, [25]). In the former, the authors presented a first naive implementation of RDD based on SPARQL 1.0, which presented an acceptable overhead in the dataset loading process and some limitations with respect to constraint types. Moreover, based on SPARQL 1.1, insights for overcoming these limitations are presented. In the case of DSP, the authors describe a framework where DSP specifications are translated into SPARQL queries, via the SPARQL Inferencing Notation (SPIN), to validate RDF data against constraints such as cardinality, value ranges, and vocabulary usage. Finally, ASP could be used for the implementation of CLs which aims to validate PGs ([147]).

5.3.4. CL applied cases

Since the definition of CLs that follows the concept of shape graphs, such as ShEx and SHACL, several models that are represented in SW technologies are validated by them. The authors found several works that presented the validation of a KRM through the application of a shape CL, which allows to validate it following CWA and UNA configurations. ShEx could be used to describe and validate existing models related to a determined domain (such as medicine) or even to validate linked data portals ([164], [97]). In the same way, SHACL is often employed for similar objectives, such as the definition of constraints and the posterior validation of domain related ontologies ([73], [85], [108], [118], [88], [157], [94], [53], [27]). SHACL is also used for the validation of other models such as domain related KGs ([154], [176], [159], [63]), already published shared vocabularies ([165]) and even systems that are not initially semantically structured on a model, such as Github repositories ([106]). In addition, SHACL validation is applied to accomplish specific functionalities such as resource access management ([171], [160]), user access authorization ([27]), product configurations ([71], [144]) and compliance checking ([118], [136]). The important amount of papers (22) inside this classification shows the high acceptance that shapes graphs (mostly represented in SHACL) are experiencing for practical applications over heterogeneous case studies.

5.3.5. CL recursion

As mentioned before, SHACL is the CL defined as the standard by the W3C. However, in the definition of the language semantics, there are some characteristics that were left open for being defined and developed in the implementations. One of these missing definitions is the recursive shapes, which describe a reference cycle between shapes. The design of SHACL creates a favorable environment for this type of constraints which reference another constraint ([34]), and for this reason, this gap has awoken the interest of the community, and some trends have appeared. The definition of recursive semantics for SHACL shapes was presented in some papers, inspired by different concepts, such as the AFT, Second order logic, logic programs and ASP ([34], [19], [120], [30], [11]). Furthermore, a general analysis of the recursion in SHACL could be found ([119]), as well as more specific analysis related to specific features such as the satisfiability or disjointment of SHACL in the presence of recursion ([120], [21]). Finally, the study of the validation task over recursive shapes have place and comes along with the proposal of new algorithms that are able to perform the task ([33], [7]). A total of 9 papers from the set of selected articles was classified as part of this category.

5.3.6. CL satisfiability and containment

Some works (5 papers from the selected group) have dedicated special efforts to the analysis of these two particular decision problems of CLs. Given a particular SHACL document, satisfiability is the problem of deciding whether there is an RDF graph which is validated by the document. Meanwhile, containment studies whether a particular SHACL document is subsumed by a second one; that is, whether all graphs that are validated by the first are also validated by the second. The study of both characteristics opens interesting paths for the improvement of the state of the art validation techniques. Some proposals were presented for the study of one of these problems. A sound and complete procedure for containment in ShEx, considering several SPARQL fragments, is presented in [2], meanwhile the same problem is studied in ShEx from a theoretical point of view in [156]. Furthermore, the analysis of satisfiability and containment in SHACL could be done through a previous translation to FOL ([121]) or a translation to a fragment of monadic second order logic which includes the recursive fragments ([120]). Moreover, the decidability of shapes containment in SHACL has been studied through a rewriting to DLs with the use of subsumption [101]. Finally, a checkpoint of the work done with respect to SHACL is presented as a review in [119].

5.3.7. CL further characteristics

The study of the CL characteristics includes more than recursion, containment and satisfiability. In this category, those studies that analyzed other features apart from the mentioned ones are presented, which represent 4 papers from the total selected ones. One characteristic that is taken into account is the SHACL arbitrary and unrestricted negation including the study of the corresponding semantics and the validation of shape graphs which includes this type of semantic ([34], [7]). Furthermore, the analysis of the SHACL semantics related to other four features is presented in one specific work, aiming to analyze their impact in the expressiveness of the language. These features are equality, disjointness, closure constraints and zero-or-one paths ([21]). Finally, the decidability over filter features, and a proposal for their effective treatment is presented in [120].

5.3.8. *Mapping techniques*

This category includes those papers in which a mapping technique is applied, which means a syntactic and semantic translation from a source language to the most possible equivalent representation in another target language. Only those works in which the source language and/or the target language is able to represent formal SCs are taken into account.

An important amount of these works is dedicated to the mapping of a source language into a generic target language (which could be even a new language) and in most of these cases, the real objective is to analyze theoretical characteristics of the source language or to generate the rewriting in the most possible generic way. SHACL is involved in many related works, and it was mapped to generic languages such as new languages of FOL and second order logic ([120]), DL ([101], [20]) and logic programming ([30]). Furthermore, both SHACL and ShEx were mapped to a minimal and new intermediate language to facilitate the comparison between them ([96]). Moreover, an idea of generalization based on the transformation of a generic high level language to a generic CL was presented in [79], with the objective of paving the way for the transformations independently of the specific source language. Finally, the transformation of SQL data to generic shapes is presented in [23].

Another trend well established is based on finding an equivalence between the SQL constraints and their representation as shape constraints. There are different approaches to this kind of mapping, such as the transformation of SQL constraints to a generic representation of shapes ([23]), or the mapping from the SQL schema to the target standard shape CL SHACL ([162], [163]). Furthermore, taking into account that RML ([16]) is a rule language that is used to map SQL data to RDF, an approach for mapping the constraints directly from these rules to SHACL shapes was also presented ([43]).

Going forward, many works have proposed mapping techniques for determining equivalences between shape constraints and the SPARQL query language, allowing to have an executable version. Going to the approaches, the mapping could be done from a generic CL to SPIN-SPARQL as an attempt to generalize this kind of translation ([79]). In addition, some techniques were presented for mapping specific constraints represented in a language to their respective representation in SPARQL queries, having as the source languages such as SHACL ([119]), RDD ([59]), DSP ([25]), Wikidata property constraints ([56]) and a combination of OWL language with SWRL rules ([150]). Moreover, the mapping could be done in the opposite way, having SPARQL as the source and a CL such as SHACL as the target ([102]).

Another type of mapping that is often used is between a CL such as SHACL and the ASP semantics, aiming to have the possibility of analyzing the constraints on the well-founded semantics of the target ([144], [11], [6]). The mapping could be also done having a SW model as a source, such as RDF KGs ([152], [114], [50]), OWL family ontologies ([127], [142], [31], [157]) or even the specific Wikidata model ([56]).

There are still more source types which are mapped to SHACL such as Object Constraint Language (OCL, [98]), DECLARE templates (a well-established declarative process specification language, [45]), and even trained decision trees ([50]) or semantic profiles generated by the ABSTAT tool ([153]). Finally, there are also two more source types which are mapped to ShEx constraints, which are XML Schema ([64]) and CSV ([140]).

Most of these techniques ease the interoperability and integration between heterogeneous models and the SW. The important amount of works classified in this category (31 papers), could be interpreted as the popularity of the SW technologies, and how many communities present a tendency to move to the latter side.

5.3.9. *SPARQL constraint implementation*

An important amount of papers (17) presented the validation of SW models through the design and execution of SPARQL queries and its use as a CL. It's worth to mention that this category does not include those implementations which merely describes a specific CL such as SHACL. The motivation for the use of SPARQL queries in order to check SCs with a closed interpretation is related to the importance and prevalence that this language has had since the beginning of the SW and also to the variety of constraints that can be expressed with this language. SPARQL constraints could be defined and implemented to validate a specific domain ontology ([115], [53]), but there are also some proposals which define the queries in a more generic way with the intention of being reusable for validating several ontologies and to reflect theoretical extensions ([151], [161], [92], [78], [107], [150]). SPARQL queries could also be related to KGs, and the objectives could be the validation of a specific KG such as Wikidata ([167], [56]), the definition of queries in a reusable generic way ([99], [123]) and even for determining semantically correct property value assertions of instance-property pairs ([46], [47]). Finally, the definition and implementation of the SPARQL queries are also done through the use of the SPARQL Inferencing Notation (SPIN) framework, and its applied with different already mentioned objectives such as the validation of ontologies ([62], [79]) and KGs ([81]) in a generic way.

It is important to note, however, that the use of SPARQL for constraint validation comes with certain limitations, especially when it comes to recursion. SPARQL 1.1 introduced property paths, which allow expressing simple recursive patterns (e.g., transitive closure via * and + operators). However, these constructs are not expressive enough to capture general recursive constraints, such as arbitrary forms of ancestor–descendant relationships, which require Datalog-like recursion beyond standard SPARQL capabilities ([131],[93]). The effectiveness of recursive constraint evaluation is also highly store-dependent. Benchmarks show that some RDF stores (e.g., Virtuoso[7] and Apache Jena[8]) differ in their completeness and efficiency when handling property paths: Virtuoso performs well on some queries but may fail on complex recursive evaluations, while Jena tends to offer more complete but sometimes slower responses ([145]). This variability highlights the importance of evaluating recursive constraint implementations across multiple stores.

5.3.10. Definition of specific types of constraints

An interesting amount of papers (21) presented in their work, an analysis and/or definition of the expressiveness of specific SCs and their validation characteristics. This category puts the focus mainly in the type of the SCs, their semantics and their expressiveness. These works could be presented in different ways, and to start describing one of them, there exist a type of work in which the attention is focused in a few SCs (less than 10) and where the definition of these constraints is done independently of the data ([99], [151], [161], [9], [115], [150], [107], [4], [116], [133], [32]). Another type of work that could be found is also related to the definition of a few constraints, but in contrary to the latter mentioned, in this case, the constraints are defined through an extraction of patterns from the data ([112], [113], [114], [56]). Finally, there are works in which the analysis or definition of several specific constraints is presented, and their support by the existing constraint representation approaches and frameworks are reviewed ([78], [79], [82], [81], [166]). These kind of works shows how linked are the validation of KRMs and the definition of SCs. Furthermore, these works correspond to the attempt of being able to represent, step by step, more sophisticated restrictions, allowing the validation in a more efficient way.

5.3.11. Schema extraction

Schema extraction includes those works which proposed a technique for the extraction of the schema (or a part of it) that is implicitly present in a KR model. This extraction is done to obtain a screenshot of the schema at one desired point in time, which will then be used for the validation of the model, between other interesting functionalities, such as supervising the evolution of the model. Having the possibility of extracting these validating schemas in an automatic way has an interesting potential, since the manual creation of these is a time-consuming and error-prone task.

To start with the enumeration of the known approaches, those techniques which extract constraints from RDF graphs are described. A technique for extracting ShEx shapes (and also OWL axioms) is based in a semi-automatic process which exploits structural statistics that then are used in a simplified process ([38]). In addition, a similar technique for extracting SHACL shapes was explained in [127]. Furthermore, the need for the development of an automatic technique for the extraction of specific structural SCs was already analyzed and proposed in [112]. There are also techniques which extract automatically a few specific constraints from these graphs and represent them as ShEx shapes, through a simulation-based approach ([68]). Finally, having as the origin a domain-specific RDF graph, a semi-automatic and database-driven technique for the extraction of SHACL shapes is presented in [139].

Moreover, the automatic extraction of certain constraints could also be done from a KB. An automatic extraction of cardinality constraints represented in ShEx was presented, based on mining the relation cardinality bounds by prior normalizing and filtering the data ([113], [114]).

Going forward, the constraints could be also extracted from ontologies. SHACL shapes could be obtained from a domain-specific ontology as in [157]. Moreover, the extraction could be generalized to any ontology as presented in [31], where the technique consists in the execution of mappings between ontology and SHACL patterns previously loaded in a KG.

Finally, the constraints could be also extracted from KGs, and some techniques reflect this possibility. A technique for the automatic induction of generic RDF shape graphs based on machine learning and data profiling was presented

---

[7] `https://virtuoso.openlinksw.com/`

[8] `https://jena.apache.org/`

in [111]. Furthermore, SHACL shapes could be extracted from KGs through the design of an algorithm based on approximation and summarization techniques ([132]). In addition, an automatic extraction of SHACL and ShEx shapes based on mining the graph structure was presented in ([55]), offering as well the possibility for inter-linkage between the generated shapes. Finally, a proposal for the automatic extraction of SHACL shapes based on Inverse Open Paths was presented in [117].

The presence and development of these sophisticated and complex techniques represent how valuable the validation of KR models currently is. In addition, it shows that it is becoming a good practice on the SW, to be able to validate models that were already published.

5.3.12. Visual constraints creation and validation

The representation and manipulation of SCs require background knowledge related to the semantic and syntax of specific CLs designed for these purposes. In order to increase the adoption of the SW technologies and the existing languages designed for validation purposes, some papers (8) present the design and development of an intermediate graphic layer between the user and the language in order to ease its use. There exist web interfaces which were designed for the creation and edition of SHACL shapes aiming to ease the validation of previously loaded ontologies ([129], [148]). Moreover, a similar tool was presented in [172] as a form based web application which allows the creation and edition of SHACL shapes, but in this case aiming to validate KGs instead. Going forward and having into account the popularity of the Protégé tool for the creation of KR models, a plug-in for the creation and the execution of these kind of shapes was presented in [52]. Furthermore, there exist tools which aim to ease the creation of ShEx shapes, such as ShExStatements, that proposes using tabular format or CSV file format for easing the task ([140]), and Validata, a web tool which is able to validate an RDF graph against these constraints, allowing in addition to define different validation requirement levels. Finally, there are works which are not only proposing visual tools for the creation and edition of SHACL shapes, but also defines baselines as an attempt to standardize the development of these kind of tools, such as the desired features that they should cover ([41]) and visual notations that could represent efficiently each kind of constraint ([103]).

5.3.13. Validation reports

This subsection focuses on the outputs generated as a result of constraint validation, particularly the structure, semantics, and potential extensions of validation reports. While these outputs may support error detection and correction, the emphasis here lies specifically on how validation results are represented and interpreted. The validation of shape constraints generates as an outcome a report with some details related to what happened during the process. The deepness and level of detail of these reports depend on the specific implementation, and presents an important potential for post-validation actions. The study of these reports was found only in 3 works between the selected articles. A rules-based reasoning technique which gives the possibility for extending validation reports with more detailed root-cause explanations, and having an accurate number of violations was presented in [40]. Moreover, the problem of the explanation of non-validation of constraints was formalized in [5], where the authors also proposed presenting a set of RDF graph updates in the report, as a guide to repair the violations found. Another type of extension found was based on adding probabilistic information in the reports, expressing a set of metrics including the generality and likelihood of shapes ([54]).

5.3.14. Error treatment

In contrast to validation reporting, this subsection addresses the broader tasks of error detection, correction, and prevention enabled by semantic constraints. These approaches often build on validation outcomes but may also function independently, aiming to improve the overall quality, maintenance, and evolution of knowledge representation models. There is evidence of these types of actions in 10 of the selected papers. These works are aimed to improve the quality, maintenance and evolution of KRMs through the exploitation of the SCs represented.

Some techniques are exclusively dedicated to error detection, as in [60], where a technique for detecting erroneous numerical values in a linked data repository is presented, paying special attention to the choice of sub populations of values and cross-checking by means of a second set of data. Another proposal for error detection in linked data is based in an automatic test instantiation based on schema constraints or semi-automatically enriched schemata, allowing users to generate their specific tests ([92]). In addition, the error detection could be also focused in relation assertions of a KG, with a technique based on a combination of path and type usage ([109]).

Going forward, there exist techniques that not only detect the errors, but also propose alternatives for the correction of the found inconsistencies. The last mentioned technique for error detection has an extension in which it includes the automatic correction of the errors, relying on type predictors and on a classifier for every relation in the graph, exploiting local feature selection ([50]). Another work presented a proposal which begins from a detail of violations presented in a SHACL validation report, codifies the problem in ASP and generates the corresponding repairs ([6]). Furthermore, these repairs could be also presented directly in the validation report, constructing the explanation using the notion of a repair as a collection of additions and deletions whose application results in a repaired graph that does satisfy the given SHACL constraints ([5]).

Finally, there exist techniques which aim to the prevention of the errors. In [175], the presented technique first finds the key instances of the data graph, then detects possible inconsistencies that could be generated after the inference of the model and finally provides the corresponding repairs to avoid these errors. A similar technique is applied when updating RDF graphs, avoiding the identified possible errors through additional updates represented as bnodes ([72]). Moreover, the error prevention could also be done in KGs through an initial evaluation of the semantically correct property value assertions, which then is used to provide a list of acceptable statements to the users. In this way, the errors are directly avoided before they are actually materialized ([46], [47]).

5.3.15. Inference in validation

The inference of SCs could be applied in KRMs with goals such as analyzing the validation procedure, improving it or even creating a new one. The selected articles include 6 proposals which involve and relate the inference process with the validation one.

Having an RDF graph and a set of SCs, an interesting research consisted in analyzing the validation process by first validating the graph as it is, and second, applying a prior inference process in order to include the implicit knowledge in a new validation of the model. The comparison between the two scenarios could include the quality of the generated data, the performance of the validation task, and the dependence of the results with the underlying semantics ([24]).

A similar analysis was presented, having an RDF graph which is consistent with a defined set of constraints as the initial condition. In these cases, the idea is to apply inference rules to the model, and then validate it a second time against the same set of constraints, in order to determine if the consistency holds. Moreover, if the second validation returns violations, a repair is possible, either deleting the conflicting inference rules or modifying the set of constraints ([122], [175]).

The utility of the inference process has been also researched specifically with respect to the explanation of non-validation results in SHACL. In this technique, the reasoning procedures are defined to decide if there are possible explanations, to find the best candidate and to decide if an explanation is relevant as an addition or deletion to the graph ([5]).

Furthermore, rule-based inferences were presented as the core of a new validation approach proposal in [40]. Taking advantage of the customizable inferencing steps, this technique explains the root cause violations in detail and generates an accurate number of the inconsistencies found.

Finally, an approach to inference shape graphs based on grammatical inference was presented in [68]. This technique proposes a simulation relation on the nodes of the input graph for identifying subtypes, and presents as an outcome, a schema that could be used to validate the input graph.

5.3.16. Validation of SCs for QO

This category describes works which develop techniques for QO. These techniques, in some part of their procedure, are related to a validation or satisfiability check of a query (or query results) against a schema (shape graph) or a group of constraints (defined with SPARQL queries). The selected group of papers includes 5 related techniques.

The optimization of the queries could be faced from the construction of a CL. Following this, a semantic extension to RDF was presented in [99], which includes specific types of constraints that are able to represent general constraints and key information from RDBs. This approach aims to minimize the lost information when mapping from RDBs to the extended RDF model, and allows to semantically optimize SPARQL queries based on constraints-implied knowledge. Furthermore, there is another related approach ([59]) in which the RDD CL is involved and the validation process through SPARQL queries is studied. Starting from this point and aiming to improve the performance of the constraint checker, the query optimization is treated in different ways, such as intra-query (individual query), intra-constraint (individual constraint) and inter-constraint (set of constraints) processes.

Going forward, the optimization process could aim to reduce the execution cost of the SPARQL queries by finding the best possible order execution. Based in a first validation of the basic graph patterns against a previously defined ShEx schema, the subjects of the triple patterns are linked to the shapes to which they could belong. This relation aims to rank the patterns depending on how many shapes are attached and then use this ranking for the definition of the order execution ([1], [3]).

Finally, an approach proposed to improve the performance of SPARQL queries through an analysis of their containment relation, having into account a previously defined ShEx schema. The containment relation between the queries is computed before their execution in the data. The procedure is based first on ShEx validators to compute the containment between two queries taking into account the schema and second, in query containment solvers to check the outcome of the first step without taking into account the ShEx constraints ([2]).

Query language and types of queries. Across these works, SPARQL is the primary query language used for semantic constraint-aware optimization, sometimes combined with constraint languages such as RDD ([59]) or ShEx ([1], [3], [2]). The types of queries that benefit from semantic constraint validation include:

- Selection and join queries (conjunctive queries), where constraints enable operator replacement, removal of redundant joins, and join reordering to reduce cost and improve efficiency ([99], [59], [1], [2], [3]).
- Aggregate queries, including grouping operations, where constraint knowledge supports more efficient evaluation ([59]).
- Complex queries with AND-OPT and UNION operators, where containment analysis under ShEx constraints avoids redundant or unsatisfiable query execution (e.g., [2]).

It is worth noting that recursive queries are not directly addressed in these works; the focus lies on conjunctive, aggregate, and containment-based optimizations. Furthermore, while RDD and ShEx are the main constraint languages explored, it would be valuable for future research to investigate how other widely adopted languages such as SHACL could support similar optimization strategies.

Potential trade-offs. While these approaches demonstrate that semantic constraints can substantially improve query performance, they also highlight important trade-offs between optimization gains and resource costs. For example, Lausen et al. [99] show that constraint-driven join reordering and elimination of redundant joins can reduce execution time, but the static analysis required to exploit constraints may increase query compilation overhead. Similarly, Fischer et al. [59] note that intra- and inter-constraint validation strategies with RDDs can reduce redundant computation but may introduce additional complexity and memory consumption when managing large sets of constraints. Abbas et al. [1, 3, 2] demonstrate that ShEx-based reordering can lower intermediate result sizes and improve execution efficiency in SPARQL engines like SPARQLGX, but this optimization may require more sophisticated schema analysis and can face scalability challenges as schema size and query complexity grow. Across these works, consistency and completeness of results are generally preserved, but the trade-off lies in balancing optimization overhead against execution efficiency. This suggests that future research should investigate adaptive strategies that can dynamically decide whether constraint-based optimizations are cost-effective for a given workload.

5.3.17. Validation algorithms

This category describes those works (9 from the selected papers) which present or analyses an algorithm for the validation of a data graph against a set of constraints. The validation process is still nowadays a very challenging task, and more related proposals are expected in the following years.

The authors in [34] presented an algorithm for the validation of an RDF graph against a set of SHACL shapes which consists in two steps. First, a minimal fixed point is calculated, focusing on all the constraints enforced by the graph. Second, if the validation hasn't been defined yet, an extension is done by assigning the shapes to a node and its successors. Furthermore, this work has been extended with two algorithms prepared for being executed in a SPARQL endpoint ([33]). The first one uses the answers of a set of SPARQL queries to build a set of propositional formulas that are processed by an SAT solver. Meanwhile, the second one performs the necessary inference on the fly and works for recursive, but tractable fragments of SHACL. Both algorithms handle validation for total or partial assignments. Going forward with SHACL shapes, an algorithm for an incremental validation of a KG which switches between data retrieving and constraint validation was presented in [58]. In this work, the proposal selects the traversal shape plans and rewrites the target and constraint queries to the fast detection of invalid entities, allowing to reduce the number of constraints that need to be checked. In addition, an attempt to validate full SHACL was based on Magic Sets (from logic programming), aiming to validate only the relevant neighborhood of the targeted nodes. In case the hard validation with the original schema fails, a lazy and target-oriented validation is done, taking into account a transformed "magic" version of the same schema ([7]).

This kind of works was also based on the validation of ShEx schemas. Four different ShEx implementations are mentioned and analyzed with their respective validation algorithms in [128]. Furthermore, a proposal for partial validation (only a part of the graph) was done for multi-type and single-type semantics. Based in the construction of a pre-typing and its minimal valid extension, these algorithms are a modified graph flooding example that verifies that the nodes satisfy their respective types. In addition, for single-type semantics, the process prevents assigning two different types to the same node ([155]). Going forward, a derivative algorithm designed for checking the same sources was presented in [65]. In this case, once a regular shape expression matches a set of triples, the derivatives are computed for each of them, without the necessity for decompose the graph or do backtracking. Finally, two algorithms were presented in [22], having the refinement algorithm that calculates the maximal typing defined as the union of all correct typings by iteratively removing unsatisfied node-type associations until a fixed point is reached. The second presented algorithm is recursive and aims to visit only a sufficiently large portion of the defined typings.

A different approach based on a previously built ontology of SCs was presented in [104]. A set of flags defined in the ontology, in addition to a set of SWRL rules, is exploited by the algorithm in order to complete the respective rules and to control the validation over the flagged instances.

5.3.18. Other approaches

This category encompasses diverse approaches not covered in previous sections, grouped for simplicity.

- The authors in [173] introduced a framework using ontological constraints from OWL-S and OWL to create semantic test cases for web service robustness testing.
- An RDF dataset validation approach utilizing OWL ontologies and SWRL rules to derive SCs from validation failures was presented in [104]. This approach maintains the OWA in the modelling and validates the dataset following a CWA. Similarly, the authors in [124] proposed marking certain concepts as complete using DBox predicates for validation, treating marked concepts as ICs and non-marked ones as incomplete.
- A framework for evaluating the relationship between RDF data graphs and schemas using structuredness functions was proposed in [13]. A related approach ([89]) utilized metrics and type/class profiles for a similar validation.
- A method for updating RDF databases with RDB characteristics using bnodes for consistency was developed in [72].
- A completeness analysis for evolving Knowledge Bases (KBs) using data profiling features to predict correct ICs and detect consistency issues was presented in [133].
- The integration of logical constraints from ontologies with machine learning to enforce DL satisfiability for consistent predictions ([135]).
- The definition of frameworks for consistency checking for Ontology-Based Data Access (OBDA) and Management (OBDM) systems, respectively, incorporating ICs ([116] and [32]).

5.4. SCs and their representation (RQ3 and RQ4)

In the first part of this section, the analysis of the CLs that were used for representing the SCs is presented. As expected, the most used language by far is the popular and standard CL SHACL, including subsets and extensions of the core language. It is followed by the ShEx CL, including extensions also in this case. Going forward and not very far from the latter one, the representations of CWA interpretations of the OWL family and SPARQL queries complete the top positions. The figure 9 shows how many times each CL has been used in the selected papers.

The second part of this section aims to describe which SCs are being represented. The authors found that not all the works describe specific constraints, and sometimes, they work over all the constraints that are related to a determined CL, or simply, the authors do not specify if they are restricting their work to some specific constraints of the specific

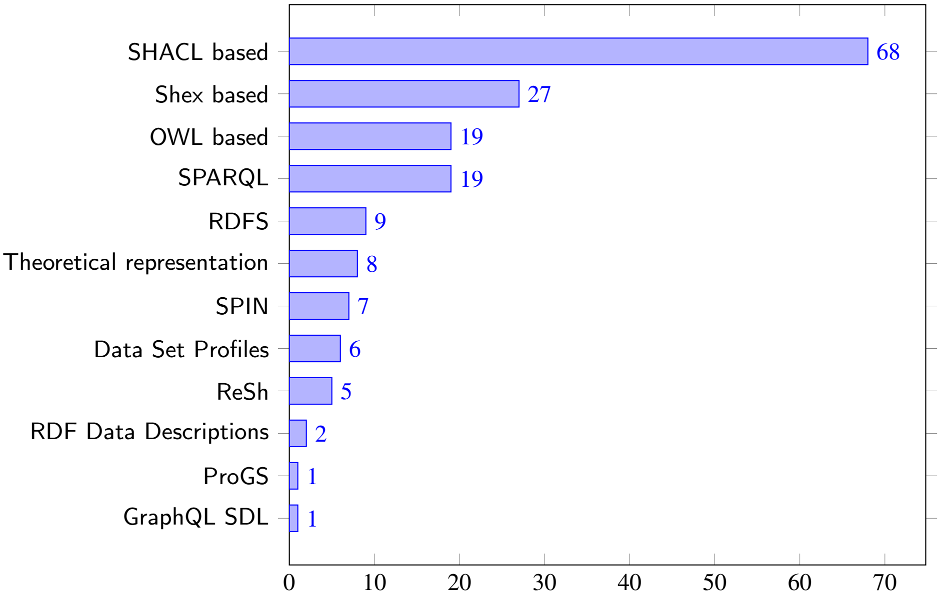

Figure 9: CLs used in the SLR selected papers

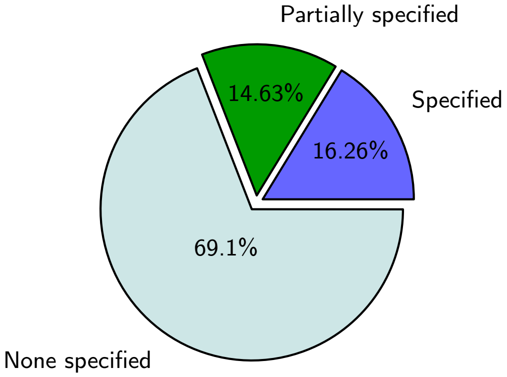

Figure 10: SLR selected papers with specified constraints

language used. In addition, there are proposals in which the constraints are partially specified. The percentage of works which describe specific constraints is shown in the figure 10.

As shown in the chart, most of the selected papers do not specify the type of constraints with which they work, and instead, the proposal is based in a CL. However, since almost a third part of the set of papers is specifying totally or partially the constraints, it's worth to analyse what are the SCs that are being highlighted.

Most of the SCs are enumerated using as a reference the several constraint types defined in [80]. The 'Data Model Consistency' constraints (also known as ICs) are one of the SCs that were used the most with 11 apparitions, what could be related to those works that were defined before the popularization of the CLs ([40], [151], [161], [9], [124], [92], [112], [32], [116], [163], [153]). This relation is made since, as shown in the set of selected papers, the first attempts to validate the consistency of knowledge formalization models were based on the definition of ICs that allowed to check the consistency by following the CWA interpretation. The 'Recursive' constraints were the second type with more presence, which is partly a consequence of the lack of definition for them in the standard specification of SHACL ([34], [30], [19], [33], [120], [147], [119], [7], [22], [142], [11]). The podium is completed by constraints that have been treated in 5 different papers, which are 'Maximum Qualified Cardinality', 'Minimum Qualified Cardinality' (both represented in [114], [111], [153], [113], [112]) and 'Property Assertion' restrictions ([50], [46], [47], [107],

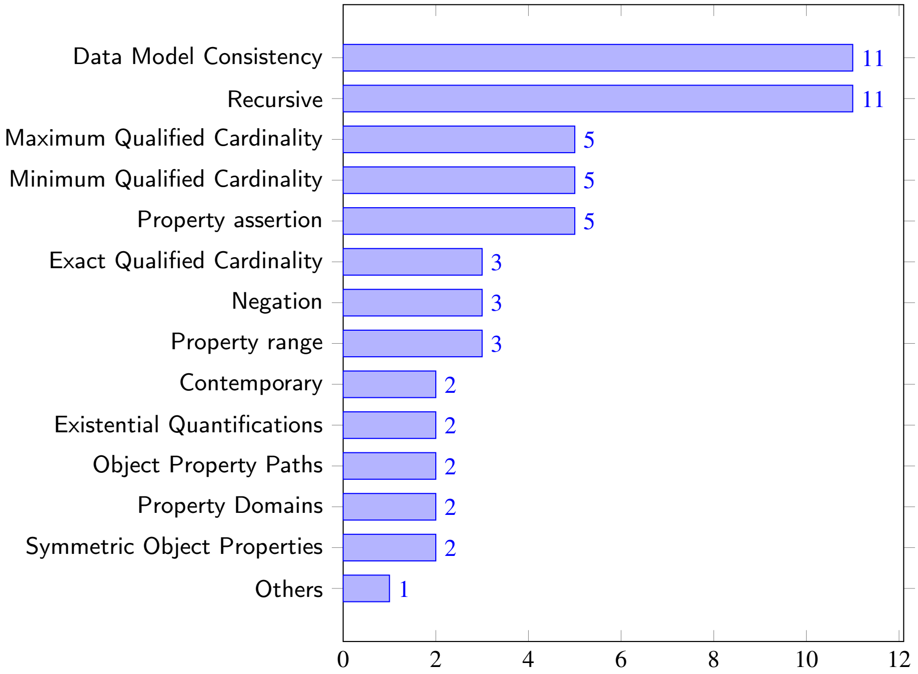

Figure 11: Specified constraints in the SLR selected papers

[60]). These constraints are frequently used for the definition of important schema characteristics. Going forward, 3 different articles presented proposals related to 'Exact Qualified Cardinality' ([114], [153], [113]), 'Negation' ([30], [7], [22]) and 'Property range' ([111], [112], [56]) restrictions, meanwhile 2 examples were found for 'Existential Quantifications' ([60], [115]), 'Object Property Paths' ([109], [21]), 'Property Domains' ([112], [56]), 'Symmetric Object Properties' ([21], [56]) and 'Contemporary' ([4], [56]). The latter constraint is not defined in the referenced classification, and instead, this constraint was defined in one of the mentioned articles ([4]). Furthermore, a list of constraints which were found only once in the selected papers is detailed as follows: constraints defined in the reference classification such as 'Allowed values' and 'Universal Quantifications' ([60]), 'Asymmetric Object Properties' ([21]), 'Class-Specific Property range', 'IRI Pattern Matching' and 'Language Tag Matching' ([56]), 'Context-Specific Property Groups' ([89]), 'Disjoint classes', 'Disjoint properties' and 'Dependency' constraint ([115]), 'Literal Pattern Matching' ([112]), 'Primary Key Properties' ([163]), 'Subsumption', 'Descriptive' and 'Prescriptive' constraints defined in [107], 'User-Defined' restrictions ([72]) and finally constraints adapted to the PG data model such as 'Comparison', 'Exact Qualified Cardinality', 'Maximum Qualified Cardinality', 'Minimum Qualified Cardinality Restrictions', 'Property specific key - edge path', 'Property specific key - node path' and 'Recursive' ([147]). The specified constraints and their distribution detail are represented in the figure 11. For a matter of simplicity and clarity, the SCs with only 1 related papers will be comprehended in the category 'Others'.

6. Discussion

In this section, the relation between the different results presented in the previous section is analysed and interpreted. In addition, the open challenges that are derived from the results are described.

6.1. Integrated results

The applications discussed in Section 5.3 are pivotal for this paper's contribution analysis. This section examines how contribution types intersect with publication years to identify current trends. Trends are determined by categories where over half of the papers were published in the last six years (from 2023 to 2018) and at least 5 papers represent this percentage.

- 90% of papers in the 'CLs - Applied cases' category were published within this timeframe, indicating a trend possibly influenced by SHACL standardization in 2017 and increased adoption for system validation based on KRMs.
- 100% of papers concerning recursion in shape graphs ('CLs - Recursion') were presented during this period, demonstrating a community effort to extend shapes CLs' features and expressiveness since the SHACL specification in 2017.
- Around 83% of works addressing satisfiability and containment of shapes were published in the last 6 years, reflecting sustained interest since SHACL's inception.
- Similarly, 83% of validation studies involving inference were released in recent years, aiming to enhance SW validation techniques by leveraging inference capabilities.
- Over 80% of papers introducing mapping techniques for contributions were published in the defined period, suggesting a growing interest in representing and validating heterogeneous sources with SW technologies.
- 75% of articles proposing visual representation and editing of shape constraints were published in recent years, indicating efforts to facilitate mass adoption of these technologies.
- Techniques for automatic or semi-automatic extraction of shape constraints from SW models saw a 69% publication rate since 2018, showcasing efforts to streamline schema creation for improved model validation.
- Nearly 65% of papers presenting CL definitions were published in the last six years, reflecting ongoing efforts to expand CL capabilities and features incrementally.
- 60% of works related to error detection, correction, and prevention were published within the defined period, suggesting a shift towards more comprehensive processes beyond validation alone.

6.2. Current challenges

The validation of SCs in KRMs for the SW is increasingly tied to the development of CLs, specially since the establishment of SHACL as a standard. However, significant research gaps exist in this emerging field. Key challenges include:

- **Visual Interfaces:** To broaden the adoption of SW technologies, there is a need for user-friendly visual interfaces for handling CLs. Few proposals currently exist, but they are in early stages.
- **Validation Reports:** SHACL standardization did not define how validation reports should be generated and shared. Future work is expected to focus on enhancing the quality and content of these reports, potentially including recommended actions such as data updates.
- **Mapping Techniques:** Efforts to develop generic mapping techniques, independent of specific contexts, are crucial for interoperability and integration with SW. In addition, more work has to be done over these techniques in order to maintain as much semantics are possible.
- **Schema Extraction Techniques:** Automating shape graph generation from KGs, ontologies, and other KR models requires more efficient methods, including addressing performance issues with large knowledge models and ensuring constraints cover diverse CLs. Moreover, the generation of shapes of spurious data and the algorithms to avoid it represent another research challenge. In addition, techniques for the extraction of shapes from PGs could be studied. Finally, the extraction of multiple schemas from heterogeneous sources and their posterior integration represents a key step.
- **Shapes Containment and Recursion:** These features could be part of new techniques for schema extraction and rewriting, potentially enhancing performance and quality. In addition, the specification of SHACL could be extended with a standard definition of these features.

- **Extensions and Analysis of CLs:** Soft validation of constraint models offers a customizable approach to dealing with the inherent incompleteness of KRMs. Combining validation with error detection, correction, and prevention presents avenues for improving model quality beyond mere inconsistency detection. It is expected that more features will be added either as extensions to existing CLs or as part of the creation of new ones.
- **Validation of PGs:** This area, beyond the SW, is nascent, with only a couple of proposals for semantic definitions. Anticipated future research will explore well-founded semantics and validation approaches.

In summary, the validation of SCs in KRMs presents various research opportunities to enhance usability, report generation, semantic definition, interoperability, automation, and error management.

6.3. Additional emerging challenges

Besides the challenges directly identified in the reviewed works, there are further aspects that, while not systematically covered in the selected literature, are nevertheless highly relevant to the future of semantic constraint validation.

6.3.1. Persistent Storage and Its Implications for Constraint Validation

The persistent storage of ontologies and knowledge graphs influences both the feasibility and efficiency of constraint validation. Native RDF stores with integrated SPARQL engines, such as Virtuoso[9] and GraphDB[10] , exhibit implementation-dependent performance characteristics, which affect how validation workloads execute in practice; surveys of SPARQL processing consistently report significant variability across systems and query features, and emphasize the role of physical design and indexing in execution efficiency [145]. In enterprise contexts, deployments may also use property-graph or hybrid storage solutions (e.g., Neo4j[11]), reflecting the broader ecosystem of knowledge graph technologies discussed in [84]. Overall, storage choices impact constraint validation in three ways: (i) query optimization opportunities depend on indexes and data layout, (ii) support for advanced query features can vary by engine (affecting which constraint-checking strategies are practical), and (iii) scalability depends on the store's execution model and memory management. These observations indicate that constraint validation should be analyzed together with the storage layer, and that comparative, system-aware evaluations remain an important direction for future work [145, 84].

A further implication concerns recursive constraints. SPARQL 1.1 property paths enable limited recursion, but not full recursive evaluation; more complex patterns typically require iterative query execution or additional reasoning support. RDF stores differ in how they handle such workloads. Virtuoso and Blazegraph[12] both implement property paths at the query engine level, with performance depending on path length and indexing strategies [145]. Apache Jena[13] combines a SPARQL engine with optional rule-based inference, which can be exploited to evaluate recursive patterns declaratively. GraphDB also supports rule-based reasoning and materialization of inferred triples, allowing certain recursive constraints to be pre-computed rather than evaluated at query time [84]. Comparative studies of SPARQL performance confirm that recursion-like queries are among the most sensitive to store-specific design decisions [145]. Hence, the feasibility and scalability of recursive constraint validation are tightly coupled with the underlying storage engine, highlighting the need for system-aware analyses of constraint languages.

6.3.2. Generation of constraints with Large Language Models (LLMs)

Recent research has begun to explore the role of pre-trained Large Language Models in constraint modelling. Michailidis et al. ([110]) demonstrate that LLMs can assist in transforming natural language problem descriptions into executable Constraint Programming (CP) models by leveraging in-context learning and retrieval-augmented prompting strategies. While this work is situated in the CP community, the underlying idea of using LLMs to bridge informal descriptions and formal constraint specifications suggests potential relevance for semantic constraint validation in knowledge graphs and ontologies. In particular, future research may investigate whether similar techniques could support the automatic generation of SHACL or ShEx shapes, or assist in formulating OWL axioms, from natural language requirements. As the SLR covers work until 2023, these developments were beyond its scope, but they highlight a promising frontier at the intersection of LLMs and semantic web technologies.

---

[9] `https://virtuoso.openlinksw.com/`

[10] `https://graphdb.ontotext.com/`

[11] `https://neo4j.com/`

[12] `https://blazegraph.com/`

[13] `https://jena.apache.org/`

7. Conclusion and future work

The validation of SCs in KRMs represents a key step for increasing the adoption of SW technologies. It is necessary to provide consistent models to ensure that the systems which are constructed on top of them, are reliable enough. In order to pave the way for realizing this objective and in order to have a clearer understanding, the authors described this problem and conducted an SLR for answering the following RQ: 'which are the existing trends related to the validation of SCs on KRMs for the SW?'. The description of the process was detailed and the results were analyzed. Based on the results, the contribution of this article included a taxonomy which described a set of categories of related research trends. Furthermore, there were two other results which were included in this taxonomy. First, the classification of the selected articles and second, the enumeration of the related current open challenges. Besides the SW models, Property Graphs are added to the analysis and results performed.

Several interesting current challenges were found and presented in this work. However, the authors are specifically interested in the potential that the schema extraction techniques bring to the community. The further steps will be based on a deep study of the related state of the art, and the design of a new technique aiming to improve not only the performance, but also the accuracy of the generated schema. The authors have already discovered that the current existing techniques present the following weakness: the generated constraints are limited in their expressiveness, the input is normally limited to one type of model, most of the outputs are represented in a specific language, the transformation is normally done in a uni-directional way, there's a lack in the treatment of the generated shapes and the performance is normally not enough for the treatment of big KGs. In addition, it is expected that the extraction of schemas could be done from heterogeneous sources, having as a result multiple shape graph that would create a scenario in which it would be interesting to integrate them into one unique schema. In consequence, further work will be done in order to propose a solution to these challenges. Moreover, the authors think about the intersection between recursive shapes, containment, shape extraction and shape integration. Including both features in the development of the new extraction and integration techniques opens an interesting path to explore.

CRediT authorship contribution statement

Mariano Julián Ferreirone: Conceptualization, Methodology, Data curation, Formal Analysis, Investigation, Writing – original draft, Writing – review & editing, Validation, Visualization. **Mario Lezoche:** Conceptualization, Methodology, Investigation, Writing – review & editing, Supervision, Project administration, Resources, Validation, Funding acquisition. **Diego Torres:** Conceptualization, Methodology, Investigation, Writing – review & editing, Supervision, Project administration, Resources, Validation, Funding acquisition. **Chiara Franciosi:** Methodology, Writing – review & editing, Validation. **Hervé Panetto:** Writing – review & editing, Resources, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding sources

This work has been funded by the Grand East Region of France (Grand Est) and by the French National Syndicate of Ski Teachers (Syndicat National des Moniteurs du Ski Français - SNMSF).

Acknowledgments

The authors thank Yoann Couble and Laurent Teste for their explanations of the limitations found in the SNMSF current data representation which were useful for comparing relational databases with knowledge representation characteristics.

References

[1] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2017a. Optimising SPARQL Query Evaluation in the Presence of ShEx Constraints, in: BDA 2017 - 33ème conférence sur la " Gestion de Données - Principes, Technologies et Applications ", Nancy, France. pp. 1–12. URL: https://inria.hal.science/hal-01673067.

[2] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2017b. Sparql query containment with shex constraints, in: Kirikova, M., Nørvåg, K., Papadopoulos, G.A. (Eds.), Advances in Databases and Information Systems, Springer International Publishing, Cham. pp. 343–356.

[3] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2018. Selectivity estimation for sparql triple patterns with shape expressions, in: Mikkonen, T., Klamma, R., Hernández, J. (Eds.), Web Engineering, Springer International Publishing, Cham. pp. 195–209. doi:https://doi.org/10.1007/978-3-319-91662-0_15.

[4] Abián, D., Bernad, J., Trillo-Lado, R., 2019. Using contemporary constraints to ensure data consistency, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Association for Computing Machinery, New York, NY, USA. p. 2303–2310. URL: https://doi.org/10.1145/3297280.3297509, doi:10.1145/3297280.3297509.

[5] Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Šimkus, M., 2021. Reasoning about Explanations for Non-validation in SHACL, in: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, pp. 12–21. URL: https://doi.org/10.24963/kr.2021/2, doi:10.24963/kr.2021/2.

[6] Ahmetaj, S., David, R., Polleres, A., Šimkus, M., 2022a. Repairing shacl constraint violations using answer set programming, in: Sattler, U., Hogan, A., Keet, M., Presutti, V., Almeida, J.P.A., Takeda, H., Monnin, P., Pirrò, G., d'Amato, C. (Eds.), The Semantic Web – ISWC 2022, Springer International Publishing, Cham. pp. 375–391. doi:https://doi.org/10.1007/978-3-031-19433-7_22.

[7] Ahmetaj, S., Löhnert, B., Ortiz, M., Šimkus, M., 2022b. Magic shapes for shacl validation. Proc. VLDB Endow. 15, 2284–2296. URL: https://doi.org/10.14778/3547305.3547329, doi:10.14778/3547305.3547329.

[8] Ait-Ameur, Y., Méry, D., 2016. Making explicit domain knowledge in formal system development. Science of Computer Programming 121, 100–127. URL: https://www.sciencedirect.com/science/article/pii/S0167642315004153, doi:https://doi.org/10.1016/j.scico.2015.12.004. special Issue on Knowledge-based Software Engineering.

[9] Akhtar, W., Cortés-Calabuig, Á., Paredaens, J., 2011. Constraints in rdf, in: Schewe, K.D., Thalheim, B. (Eds.), Semantics in Data and Knowledge Bases, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 23–39. doi:https://doi.org/10.1007/978-3-642-23441-5_2.

[10] Akkermans, H., Harmelen, F., Schreiber, G., Wielinga, B., 1992. Formalization of knowledge-level models for knowledge acquisition. International Journal of Intelligent Systems - IJIS 8. doi:10.5555/171647.171659.

[11] Andresel, M., Corman, J., Ortiz, M., Reutter, J.L., Savkovic, O., Simkus, M., 2020. Stable model semantics for recursive shacl, in: Proceedings of The Web Conference 2020, Association for Computing Machinery, New York, NY, USA. p. 1570–1580. URL: https://doi.org/10.1145/3366423.3380229, doi:10.1145/3366423.3380229.

[12] Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., Vrgoč, D., 2017. Foundations of modern query languages for graph databases. ACM Comput. Surv. 50. URL: https://doi.org/10.1145/3104031, doi:10.1145/3104031.

[13] Arenas, M., Díaz, G., Fokoue, A., Kementsietsidis, A., Srinivas, K., 2014. A principled approach to bridging the gap between graph data and their schemas. Proc. VLDB Endow. 7, 601–612. URL: https://doi.org/10.14778/2732296.2732297, doi:10.14778/2732296.2732297.

[14] Bassiliades, N., Symeonidis, M., Gouvas, P., Kontopoulos, E., Meditskos, G., Vlahavas, I., 2018. Paasport semantic model: An ontology for a platform-as-a-service semantically interoperable marketplace. Data Knowledge Engineering 113, 81–115. URL: https://www.sciencedirect.com/science/article/pii/S0169023X17300551, doi:https://doi.org/10.1016/j.datak.2017.11.001.

[15] Beimel, D., Peleg, M., 2011. Using owl and swrl to represent and reason with situation-based access control policies. Data Knowledge Engineering 70, 596–615. URL: https://www.sciencedirect.com/science/article/pii/S0169023X11000334, doi:https://doi.org/10.1016/j.datak.2011.03.006.

[16] Ben De Meester, P.H., Delva, T., 2014. Rdf mapping language (rml). URL: https://rml.io/specs/rml/.

[17] Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. Scientific American 284, 34–43. URL: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

[18] Bizer, C., Heath, T., Berners-Lee, T., 2023. Linked Data - The Story So Far. 1 ed.. Association for Computing Machinery, New York, NY, USA. p. 115–143. URL: https://doi.org/10.1145/3591366.3591378, doi:10.4018/jswis.2009081901.

[19] Bogaerts, B., Jakubowski, M., 2021. Fixpoint semantics for recursive shacl. doi:10.48550/arXiv.2109.08285.

[20] Bogaerts, B., Jakubowski, M., Van den Bussche, J., 2022. Shacl: A description logic in disguise, in: Gottlob, G., Inclezan, D., Maratea, M. (Eds.), Logic Programming and Nonmonotonic Reasoning, Springer International Publishing, Cham. pp. 75–88. doi:https://doi.org/10.1007/978-3-031-15707-3_7.

[21] Bogaerts, B., Jakubowski, M., den Bussche, J.V., 2023. Expressiveness of shacl features and extensions for full equality and disjointness tests. URL: https://arxiv.org/abs/2212.03553, doi:https://doi.org/10.46298/lmcs-20(1:16)2024, arXiv:2212.03553.

[22] Boneva, I., Labra Gayo, J.E., Prud'hommeaux, E.G., 2017. Semantics and validation of shapes schemas for rdf, in: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (Eds.), The Semantic Web – ISWC 2017, Springer International Publishing, Cham. pp. 104–120. doi:https://doi.org/10.1007/978-3-319-68288-4_7.

[23] Boneva, I., Staworko, S., Lozano, J., 2020. Consistency and certain answers in relational to rdf data exchange with shape constraints, in: Darmont, J., Novikov, B., Wrembel, R. (Eds.), New Trends in Databases and Information Systems, Springer International Publishing, Cham. pp. 97–107. doi:https://doi.org/10.1007/978-3-030-54623-6_9.

[24] Bosch, T., Acar, E., Nolle, A., Eckert, K., 2015a. The role of reasoning for rdf validation, in: Proceedings of the 11th International Conference on Semantic Systems, Association for Computing Machinery, New York, NY, USA. p. 33–40. URL: https://doi.org/10.1145/2814864.2814867, doi:10.1145/2814864.2814867.

[25] Bosch, T., Eckert, K., 2014. Towards description set profiles for rdf using sparql as intermediate language, in: Proceedings of the 2014 International Conference on Dublin Core and Metadata Applications, Dublin Core Metadata Initiative. p. 129–137. doi:10.5555/2771234.

2771248.

[26] Bosch, T., Nolle, A., Acar, E., Eckert, K., 2015b. Rdf validation requirements - evaluation and logical underpinning. URL: `https://arxiv.org/abs/1501.03933`, doi:`https://doi.org/10.48550/arXiv.1501.03933`, `arXiv:1501.03933`.

[27] Braun, C.H.J., Papanchev, V., Käfer, T., 2023. Sissi: An architecture for semantic interoperable self-sovereign identity-based access control on the web, in: Proceedings of the ACM Web Conference 2023, Association for Computing Machinery, New York, NY, USA. p. 3011–3021. URL: `https://doi.org/10.1145/3543507.3583409`, doi:`10.1145/3543507.3583409`.

[28] Brickley, D., Guha, R., 2000. Resource description framework schema (rdf/s) specification 1.1 .

[29] Chabin, J., Halfeld-Ferrari, M., Laurent, D., 2020. Consistent updating of databases with marked nulls. Knowl. Inf. Syst. 62, 1571–1609. URL: `https://doi.org/10.1007/s10115-019-01402-w`, doi:`10.1007/s10115-019-01402-w`.

[30] Chmurovic, A., Simkus, M., 2022. Well-founded semantics for recursive SHACL, in: Alviano, M., Pieris, A. (Eds.), Proceedings of the 4th International Workshop on the Resurgence of Datalog in Academia and Industry (Datalog-2.0 2022) co-located with the 16th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2022), Genova-Nervi, Italy, September 5, 2022, CEUR-WS.org. pp. 2–13. URL: `https://ceur-ws.org/Vol-3203/paper1.pdf`.

[31] Cimmino, A., Fernández-Izquierdo, A., García-Castro, R., 2020. Astrea: Automatic generation of shacl shapes from ontologies, in: Harth, A., Kirrane, S., Ngonga Ngomo, A.C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 497–513.

[32] Console, M., Lenzerini, M., 2020. Epistemic integrity constraints for ontology-based data management. Proceedings of the AAAI Conference on Artificial Intelligence 34, 2790–2797. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/5667`, doi:`10.1609/aaai.v34i03.5667`.

[33] Corman, J., Florenzano, F., Reutter, J.L., Savković, O., 2019. Validating shacl constraints over a sparql endpoint, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 145–163.

[34] Corman, J., Reutter, J.L., Savković, O., 2018. Semantics and validation of recursive shacl, in: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E. (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham. pp. 318–336.

[35] Costa, S.D., Barcellos, M.P., de Almeida Falbo, R., Conte, T., de Oliveira, K.M., 2022. A core ontology on the human–computer interaction phenomenon. Data Knowledge Engineering 138, 101977. URL: `https://www.sciencedirect.com/science/article/pii/S0169023X21000951`, doi:`https://doi.org/10.1016/j.datak.2021.101977`.

[36] Cyganiak, R., Hyland-Wood, D., Lanthaler, M., 2014. Rdf 1.1 concepts and abstract syntax. W3C Proposed Recommendation .

[37] Dalponte Ayastuy, M., Torres, D., Fernández, A., 2021. Adaptive gamification in collaborative systems, a systematic mapping study. Computer Science Review 39, 100333. URL: `https://www.sciencedirect.com/science/article/pii/S1574013720304330`, doi:`https://doi.org/10.1016/j.cosrev.2020.100333`.

[38] Dam, J., Koehorst, J., Schaap, P., Santos, V., Suarez-Diez, M., 2015. Rdf2graph a tool to recover, understand and validate the ontology of an rdf resource. Journal of biomedical semantics 6, 39. doi:`10.1186/s13326-015-0038-9`.

[39] Davis, R., Shrobe, H., Szolovits, P., 1993. What is a knowledge representation? AI Magazine 14, 17. URL: `https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1029`, doi:`10.1609/aimag.v14i1.1029`.

[40] De Meester, B., Heyvaert, P., Arndt, D., Dimou, A., Verborgh, R., 2020. Rdf graph validation using rule-based reasoning. Semantic Web 12, 117–142. doi:`10.3233/SW-200384`.

[41] De Meester, B., Heyvaert, P., Dimou, A., Verborgh, R., 2018. Towards a uniform user interface for editing data shapes, in: Ivanova, V., Lambrix, P., Lohmann, S., Pesquita, C. (Eds.), Proceedings of the 4th International Workshop on Visualization and Interaction for Ontologies and Linked Data. URL: `http://ceur-ws.org/Vol-2187/paper2.pdf`.

[42] Debruyne, C., Pandit, H.J., Lewis, D., O'Sullivan, D., 2020. "just-in-time" generation of datasets by considering structured representations of given consent for gdpr compliance. Knowl. Inf. Syst. 62, 3615–3640. URL: `https://doi.org/10.1007/s10115-020-01468-x`, doi:`10.1007/s10115-020-01468-x`.

[43] Delva, T., De Smedt, B., Min Oo, S., Van Assche, D., Lieber, S., Dimou, A., 2021. Rml2shacl: Rdf generation taking shape, in: Proceedings of the 11th Knowledge Capture Conference, Association for Computing Machinery, New York, NY, USA. p. 153–160. URL: `https://doi.org/10.1145/3460210.3493562`, doi:`10.1145/3460210.3493562`.

[44] Delva, Thomas and Dimou, Anastasia and Jakubowksi, Maxime and Van den Bussche, Jan, 2023. Data provenance for SHACL, in: Proceedings 26th International Conference on Extending Database Technology (EDBT 2023), pp. 285–297. URL: `http://doi.org/10.48786/edbt.2023.23`.

[45] Di Ciccio, C., Ekaputra, F.J., Cecconi, A., Ekelhart, A., Kiesling, E., 2019. Finding non-compliances with declarative process constraints through semantic technologies, in: Cappiello, C., Ruiz, M. (Eds.), Information Systems Engineering in Responsible Information Systems, Springer International Publishing, Cham. pp. 60–74. doi:`https://doi.org/10.1007/978-3-030-21297-1_6`.

[46] Dibowski, H., 2021. Property assertion constraints for an informed, error-preventing expansion of knowledge graphs, in: Villazón-Terrazas, B., Ortiz-Rodríguez, F., Tiwari, S., Goyal, A., Jabbar, M. (Eds.), Knowledge Graphs and Semantic Web, Springer International Publishing, Cham. pp. 234–248. doi:`https://doi.org/10.1007/978-3-030-91305-2_18`.

[47] Dibowski, H., 2022. Property assertion constraints for ontologies and knowledge graphs. Data Technologies and Applications , 1–20doi:`10.1108/DTA-05-2022-0209`.

[48] Ding, Y., Fensel, D., Klein, M., Omelayenko, B., 2002. The semantic web: yet another hip? Data Knowledge Engineering 41, 205–227. URL: `https://www.sciencedirect.com/science/article/pii/S0169023X02000411`, doi:`https://doi.org/10.1016/S0169-023X(02)00041-1`.

[49] Díaz Rodríguez, N., Cuéllar, M.P., Lilius, J., Delgado Calvo-Flores, M., 2014. A fuzzy ontology for semantic modelling and recognition of human behaviour. Knowledge-Based Systems 66, 46–60. URL: `https://www.sciencedirect.com/science/article/pii/`

S0950705114001385, doi:https://doi.org/10.1016/j.knosys.2014.04.016.

[50] d'Amato, C., Melo, A., Paulheim, H., 2020. Automatic detection of relation assertion errors and induction of relation constraints. Semant. Web 11, 801–830. URL: https://doi.org/10.3233/SW-200369, doi:10.3233/SW-200369.

[51] Ehrlinger, L., Wöß, W., 2016. Towards a definition of knowledge graphs.

[52] Ekaputra, F.J., Lin, X., 2016. Shacl4p: Shacl constraints validation within protégé ontology editor, in: 2016 International Conference on Data and Software Engineering (ICoDSE), pp. 1–6. doi:10.1109/ICODSE.2016.7936162.

[53] Faraj, G., Micsik, A., 2021. Representing and validating cultural heritage knowledge graphs in cidoc-crm ontology. Future Internet 13. URL: https://www.mdpi.com/1999-5903/13/11/277, doi:10.3390/fi13110277.

[54] Felin, R., Faron, C., Tettamanzi, A.G.B., 2023. A framework to include and exploit probabilistic information in shacl validation reports, in: Pesquita, C., Jimenez-Ruiz, E., McCusker, J., Faria, D., Dragoni, M., Dimou, A., Troncy, R., Hertling, S. (Eds.), The Semantic Web, Springer Nature Switzerland, Cham. pp. 91–104. doi:https://doi.org/10.1007/978-3-031-33455-9_6.

[55] Fernandez-Álvarez, D., Labra-Gayo, J.E., Gayo-Avello, D., 2022. Automatic extraction of shapes using shexer. Knowledge-Based Systems 238, 107975. URL: https://www.sciencedirect.com/science/article/pii/S0950705121010972, doi:https://doi.org/10.1016/j.knosys.2021.107975.

[56] Ferranti, N., P.A.D.S.J.A.S., 2022. Formalizing property constraints in wikidata URL: http://hdl.handle.net/20.500.12708/150282.

[57] Ferreirone, M., Lezoche, M., Torres, D., Panetto, H., 2023. Transformations on knowledge representation between OWL and RDF knowledge graphs: A study case. SADIO electronic journal of informatics and operations research 22, 80–93. URL: https://hal.science/hal-04312488.

[58] Figuera, M., Rohde, P.D., Vidal, M.E., 2021. Trav-shacl: Efficiently validating networks of shacl constraints, in: Proceedings of the Web Conference 2021, Association for Computing Machinery, New York, NY, USA. p. 3337–3348. URL: https://doi.org/10.1145/3442381.3449877, doi:10.1145/3442381.3449877.

[59] Fischer, P., Lausen, G., Schätzle, A., Schmidt, M., 2015. Rdf constraint checking.

[60] Fleischhacker, D., Paulheim, H., Bryl, V., Völker, J., Bizer, C., 2014. Detecting errors in numerical linked data using cross-checked outlier detection, in: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (Eds.), The Semantic Web – ISWC 2014, Springer International Publishing, Cham. pp. 357–372. doi:https://doi.org/10.1007/978-3-319-11964-9_23.

[61] Franciosi, C., Voisin, A., Miranda, S., Riemma, S., Iung, B., 2020. Measuring maintenance impacts on sustainability of manufacturing industries: from a systematic literature review to a framework proposal. Journal of Cleaner Production 260, 121065. URL: https://www.sciencedirect.com/science/article/pii/S0959652620311124, doi:https://doi.org/10.1016/j.jclepro.2020.121065.

[62] Fürber, C., Hepp, M., 2010. Using sparql and spin for data quality management on the semantic web, in: Abramowicz, W., Tolksdorf, R. (Eds.), Business Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 35–46. doi:https://doi.org/10.1007/978-3-642-12814-1_4.

[63] Gaignard, A., Rosnet, T., Lamotte, F., Lefort, V., Devignes, M.D., 2023. Fair-checker: supporting digital resource findability and reuse with knowledge graphs and semantic web standards. Journal of Biomedical Semantics 14. doi:10.1186/s13326-023-00289-5.

[64] García-González, H., Labra Gayo, J., 2018. Xmlschema2shex: Converting xml validation to rdf validation. Semantic Web , 1–19doi:10.3233/SW-180329.

[65] Gayo, J., Prud'hommeaux, E., Boneva, I., Staworko, S., Solbrig, H., Hym, S., 2015. Towards an rdf validation language based on regular expression derivatives, in: Proceedings of the Workshops of the EDBT/ICDT 2015 Joint Conference (EDBT/ICDT), Brussels, Belgium, March 27th, 2015., pp. 197–204.

[66] Gayo, J.E.L., 2022. Wshex: A language to describe and validate wikibase entities. URL: https://arxiv.org/abs/2208.02697, doi:https://doi.org/10.48550/arXiv.2208.02697, arXiv:2208.02697.

[67] Gomez-Perez, A., 1999. Ontological engineering: A state of the art. Expert Update 2.

[68] Groz, B., Lemay, A., Staworko, S., Wieczorek, P., 2022. Inference of Shape Graphs for Graph Databases, in: International Conference on Database Theory, Edinbourgh, United Kingdom. URL: https://inria.hal.science/hal-03559309, doi:10.4230/LIPIcs.ICDT.2022.7.

[69] Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220. URL: https://www.sciencedirect.com/science/article/pii/S1042814383710083, doi:https://doi.org/10.1006/knac.1993.1008.

[70] Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies 43, 907–928. URL: https://www.sciencedirect.com/science/article/pii/S1071581985710816, doi:https://doi.org/10.1006/ijhc.1995.1081.

[71] Haav, H.m., Maigre, R., 2019. A semantic model for product configuration in timber industry. doi:10.3233/978-1-61499-941-6-143.

[72] Halfeld-Ferrari, M., Hara, C.S., Uber, F.R., 2017. Rdf updates with constraints, in: Różewski, P., Lange, C. (Eds.), Knowledge Engineering and Semantic Web, Springer International Publishing, Cham. pp. 229–245. doi:https://doi.org/10.1007/978-3-319-69548-8_16.

[73] Hamdan, A.H., Scherer, R., 2020. Integration of bim-related bridge information in an ontological knowledgebase.

[74] Hansen, J., Beveridge, A., Farmer, R., Gehrmann, L., Gray, A., Khutan, S., Robertson, T., Val, J., 2015. Validata: An online tool for testing rdf data conformance.

[75] Harmelen, F., Lifschitz, V., Porter, B., 2007. The handbook of knowledge representation. Elsevier Science San Diego, USA , 1034.

[76] Hartig, O., 2014. Reconciliation of rdf* and property graphs. CoRR abs/1409.3288. URL: http://arxiv.org/abs/1409.3288, arXiv:1409.3288.

[77] Hartig, O., Hidders, J., 2019. Defining schemas for property graphs by using the graphql schema definition language, in: Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/3327964.3328495, doi:10.1145/

3327964.3328495.

[78] Hartmann, T., Eckert, K., 2014. Requirements on rdf constraint formulation and validation.

[79] Hartmann, T., Eckert, K., 2015. Guidance, please! towards a framework for rdf-based constraint languages. doi:10.5555/2907896.2907906.

[80] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2015. Constraints to validate rdf data quality on common vocabularies in the social, behavioral, and economic sciences. URL: https://arxiv.org/abs/1504.04479, doi:https://doi.org/10.48550/arXiv.1504.04479, arXiv:1504.04479.

[81] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2016a. Directing the development of constraint languages by checking constraints on rdf data. International Journal of Semantic Computing 10, 193–217. URL: https://doi.org/10.1142/S1793351X16400079, doi:10.1142/S1793351X16400079, arXiv:https://doi.org/10.1142/S1793351X16400079.

[82] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2016b. Validating rdf data quality using constraints to direct the development of constraint languages, in: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), pp. 116–123. doi:10.1109/ICSC.2016.43.

[83] Hitzler, P., 2021. A review of the semantic web field. Communications of the ACM 64, 76–83.

[84] Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A., 2021. Knowledge graphs. ACM Comput. Surv. 54. URL: https://doi.org/10.1145/3447772, doi:10.1145/3447772.

[85] Hou, Z.W., Qin, C.Z., Zhu, A.X., Wang, Y.J., Liang, P., Wang, Y.J., Zhu, Y.Q., 2021. Formalizing parameter constraints to support intelligent geoprocessing: A shacl-based method. ISPRS International Journal of Geo-Information 10. URL: https://www.mdpi.com/2220-9964/10/9/605, doi:10.3390/ijgi10090605.

[86] Jalali, S., Wohlin, C., 2012. Systematic literature studies: Database searches vs. backward snowballing, in: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 29–38. doi:10.1145/2372251.2372257.

[87] Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P., 2021. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE transactions on neural networks and learning systems PP. doi:10.1109/TNNLS.2021.3070843.

[88] K Soman, R., 2019. Modelling construction scheduling constraints using shapes constraint language (shacl), pp. 351–358. doi:10.35490/EC3.2019.170.

[89] Kellou-Menouer, K., Kedad, Z., 2015. Evaluating the gap between an rdf dataset and its schema, in: Jeusfeld, M.A., Karlapalem, K. (Eds.), Advances in Conceptual Modeling, Springer International Publishing, Cham. pp. 283–292. doi:https://doi.org/10.1007/978-3-319-25747-1_28.

[90] Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering 2.

[91] Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering – a systematic literature review. Information and Software Technology 51, 7–15. URL: https://www.sciencedirect.com/science/article/pii/S0950584908001390, doi:https://doi.org/10.1016/j.infsof.2008.09.009. special Section - Most Cited Articles in 2002 and Regular Research Papers.

[92] Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A., 2014. Test-driven evaluation of linked data quality, in: Proceedings of the 23rd International Conference on World Wide Web, Association for Computing Machinery, New York, NY, USA. p. 747–758. URL: https://doi.org/10.1145/2566486.2568002, doi:10.1145/2566486.2568002.

[93] Kostylev, E.V., Reutter, J.L., Romero, M., Vrgoč, D., 2015. Sparql with property paths, in: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S. (Eds.), The Semantic Web - ISWC 2015, Springer International Publishing, Cham. pp. 3–18.

[94] Köcher, A., da Silva, L.M.V., Fay, A., 2021. Constraint checking of skills using shacl, in: 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), pp. 1–6. doi:10.1109/INDIN45523.2021.9557549.

[95] Labra Gayo, J., Prud'hommeaux, E., Boneva, I., Kontokostas, D., 2017. Validating rdf data. Synthesis Lectures on the Semantic Web: Theory and Technology 7, 1–328. doi:10.2200/S00786ED1V01Y201707WBE016.

[96] Labra-Gayo, J.E., García-González, H., Fernández-Alvarez, D., Prud'hommeaux, E., 2019. Challenges in RDF Validation. Springer International Publishing, Cham. pp. 121–151. URL: https://doi.org/10.1007/978-3-030-06149-4_6, doi:10.1007/978-3-030-06149-4_6.

[97] Labra-Gayo, J.E., Prud'hommeaux, E., Solbrig, H., Boneva, I., 2014. Validating and describing linked data portals using shapes. URL: https://arxiv.org/abs/1701.08924, doi:https://doi.org/10.48550/arXiv.1701.08924, arXiv:1701.08924.

[98] Larhrib, M., Escribano, M., Cerrada, C., Escribano, J.J., 2020. Converting ocl and cgmes rules to shacl in smart grids. IEEE Access 8, 177255–177266. doi:10.1109/ACCESS.2020.3026941.

[99] Lausen, G., Meier, M., Schmidt, M., 2008. Sparqling constraints for rdf, in: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, Association for Computing Machinery, New York, NY, USA. p. 499–509. URL: https://doi.org/10.1145/1353343.1353404, doi:10.1145/1353343.1353404.

[100] Lei, X., Wu, P., Zhu, J., Wang, J., 2022. Ontology-based information integration: A state-of-the-art review in road asset management. Archives of Computational Methods in Engineering 29. doi:https://doi.org/10.1007/s11831-021-09668-6.

[101] Leinberger, M., Seifer, P., Rienstra, T., Lämmel, R., Staab, S., 2020. Deciding shacl shape containment through description logics reasoning, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 366–383. doi:https://doi.org/10.1007/978-3-030-62419-4_21.

[102] Leinberger, M., Seifer, P., Schon, C., Lämmel, R., Staab, S., 2019. Type checking program code using shacl, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 399–417. doi:https://doi.org/10.1007/978-3-030-30793-6_23.

[103] Lieber, S., De Meester, B., Heyvaert, P., Brückmann, F., Wambacq, R., Mannens, E., Verborgh, R., Dimou, A., 2021. Visual notations for viewing rdf constraints with unshacled. Semantic Web 13, 1–36. doi:10.3233/SW-210450.

[104] Lodemann, M., Marnau, R., Luttenberger, N., 2012. Using semantic constraints for data verification in an open world, in: Lukose, D., Ahmad, A.R., Suliman, A. (Eds.), Knowledge Technology, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 206–215.

[105] Marchetti, A., Ronzano, F., Tesconi, M., Minutoli, S., 2008. Formalizing knowledge by ontologies: Owl and kif .

[106] Martin, L., Henrich, A., 2022. Specification and validation of quality criteria for git repositories using rdf and shacl, in: Reuss, P., Eisenstadt, V., Schönborn, J., Schäfer, J. (Eds.), Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, Oktober 5-7th, 2022, RWTH Aachen, Hildesheim (Germany). pp. 124–135. URL: `https://ceur-ws.org/Vol-3341/WM-LWDA_2022_CRC_1149.pdf`. `https://fis.uni-bamberg.de/handle/uniba/89863`.

[107] Martin, P.A., 2018. Evaluating ontology completeness via sparql and relations-between-classes based constraints, in: 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), pp. 255–263. doi:`10.1109/QUATIC.2018.00045`.

[108] Martínez-Costa, C., Schulz, S., 2017. Validating ehr clinical models using ontology patterns. Journal of Biomedical Informatics 76, 124–137. URL: `https://www.sciencedirect.com/science/article/pii/S1532046417302381`, doi:`https://doi.org/10.1016/j.jbi.2017.11.001`.

[109] Melo, A., Paulheim, H., 2017. Detection of relation assertion errors in knowledge graphs, in: Proceedings of the 9th Knowledge Capture Conference, Association for Computing Machinery, New York, NY, USA. URL: `https://doi.org/10.1145/3148011.3148033`, doi:`10.1145/3148011.3148033`.

[110] Michailidis, K., Tsouros, D., Guns, T., 2024. Constraint Modelling with LLMs Using In-Context Learning, in: Shaw, P. (Ed.), 30th International Conference on Principles and Practice of Constraint Programming (CP 2024), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany. pp. 20:1–20:27. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CP.2024.20`, doi:`10.4230/LIPIcs.CP.2024.20`.

[111] Mihindukulasooriya, N., Rashid, M.R.A., Rizzo, G., García-Castro, R., Corcho, O., Torchiano, M., 2018. Rdf shape induction using knowledge base profiling, in: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, Association for Computing Machinery, New York, NY, USA. p. 1952–1959. URL: `https://doi.org/10.1145/3167132.3167341`, doi:`10.1145/3167132.3167341`.

[112] Muñoz, E., 2016. On learnability of constraints from rdf data, in: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (Eds.), The Semantic Web. Latest Advances and New Domains, Springer International Publishing, Cham. pp. 834–844. doi:`https://doi.org/10.1007/978-3-319-34129-3_52`.

[113] Muñoz, E., Nickles, M., 2017. Mining cardinalities from knowledge bases, in: Benslimane, D., Damiani, E., Grosky, W.I., Hameurlain, A., Sheth, A., Wagner, R.R. (Eds.), Database and Expert Systems Applications, Springer International Publishing, Cham. pp. 447–462. doi:`https://doi.org/10.1007/978-3-319-64468-4_34`.

[114] Muñoz, E., Nickles, M., 2018. Statistical Relation Cardinality Bounds in Knowledge Bases. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 67–97. URL: `https://doi.org/10.1007/978-3-662-58415-6_3`, doi:`10.1007/978-3-662-58415-6_3`.

[115] Nguyen, T.H.H., Le-Thanh, N., 2014. Ensuring the semantic correctness of workflow processes: an ontological approach, in: Proceedings of the 10th International Conference on Knowledge Engineering and Software Engineering - Volume 1289, CEUR-WS.org, Aachen, DEU. p. 30–41. doi:`10.5555/3053689.3053695`.

[116] Nikolaou, C., Cuenca Grau, B., Kostylev, E.V., Kaminski, M., Horrocks, I., 2019. Satisfaction and implication of integrity constraints in ontology-based data access, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 1829–1835. URL: `https://doi.org/10.24963/ijcai.2019/253`, doi:`10.24963/ijcai.2019/253`.

[117] Omran, P., Taylor, K., Rodríguez Méndez, S., Haller, A., 2023. Learning shacl shapes from knowledge graphs. Semantic Web 14, 1–21. doi:`10.3233/SW-223063`.

[118] Pandit, H.J., O'Sullivan, D., Lewis, D., 2019. Test-driven approach towards gdpr compliance, in: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (Eds.), Semantic Systems. The Power of AI and Knowledge Graphs, Springer International Publishing, Cham. pp. 19–33. doi:`https://doi.org/10.1007/978-3-030-33220-4_2`.

[119] Pareti, P., Konstantinidis, G., 2022. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. Springer International Publishing, Cham. pp. 115–144. URL: `https://doi.org/10.1007/978-3-030-95481-9_6`, doi:`10.1007/978-3-030-95481-9_6`.

[120] Pareti, P., Konstantinidis, G., Mogavero, F., 2022. Satisfiability and containment of recursive shacl. Journal of Web Semantics 74, 100721. URL: `https://www.sciencedirect.com/science/article/pii/S1570826822000130`, doi:`https://doi.org/10.1016/j.websem.2022.100721`.

[121] Pareti, P., Konstantinidis, G., Mogavero, F., Norman, T.J., 2020. Shacl satisfiability and containment, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 474–493. doi:`https://doi.org/10.48550/arXiv.2009.09806`.

[122] Pareti, P., Konstantinidis, G., Norman, T.J., Şensoy, M., 2019. Shacl constraints with inference rules, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 539–557. doi:`https://doi.org/10.1007/978-3-030-30793-6_31`.

[123] Patel-Schneider, P.F., 2015. Using description logics for rdf constraint checking and closed-world recognition, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press. p. 247–253. doi:`10.5555/2887007.2887042`.

[124] Patel-Schneider, P.F., Franconi, E., 2012. Ontology constraints in incomplete and complete data, in: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (Eds.), The Semantic Web – ISWC 2012, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 444–459. doi:`https://doi.org/10.1007/978-3-642-35176-1_28`.

[125] Pavlic, M., Meštrović, A., Jakupovic, A., 2013. Graph-based formalisms for knowledge representation.

[126] Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1–18. URL: `https://www.sciencedirect.com/science/article/pii/S0950584915000646`, doi:`https://doi.org/10.1016/j.infsof.2015.03.007`.

[127] Potoniec, J., Jakubowski, P., Ławrynowicz, A., 2017. Swift linked data miner: Mining owl 2 el class expressions directly from online rdf datasets. Journal of Web Semantics 46-47, 31–50. URL: `https://www.sciencedirect.com/science/article/pii/S157082681730032X`, doi:`https://doi.org/10.1016/j.websem.2017.08.001`.

[128] Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H., 2014. Shape expressions: an rdf validation and transformation language, in: Proceedings of the 10th International Conference on Semantic Systems, Association for Computing Machinery, New York, NY, USA. p. 32–40. URL: `https://doi.org/10.1145/2660517.2660523`, doi:`10.1145/2660517.2660523`.

[129] Publio, G.C., 2018. Shark: A test-driven framework for design and evolution of ontologies, in: Gangemi, A., Gentile, A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan, J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International Publishing, Cham. pp. 314–324. doi:`https://doi.org/10.1007/978-3-319-98192-5_50`.

[130] Pulido, J., Ruiz, M., Herrera, R., Cabello, E., Legrand, S., Elliman, D., 2006. Ontology languages for the semantic web: A never completely updated review. Knowledge-Based Systems 19, 489–497. URL: `https://www.sciencedirect.com/science/article/pii/S0950705106000736`, doi:`https://doi.org/10.1016/j.knosys.2006.04.013`. creative Systems.

[131] Pérez, J., Arenas, M., Gutierrez, C., 2010. nsparql: A navigational language for rdf. Journal of Web Semantics 8, 255–270. URL: `https://www.sciencedirect.com/science/article/pii/S157082681000003X`, doi:`https://doi.org/10.1016/j.websem.2010.01.002`. semantic Web Challenge 2009 User Interaction in Semantic Web research.

[132] Rabbani, K., Lissandrini, M., Hose, K., 2023. Extraction of validating shapes from very large knowledge graphs. Proc. VLDB Endow. 16, 1023–1032. URL: `https://doi.org/10.14778/3579075.3579078`, doi:`10.14778/3579075.3579078`.

[133] Rashid, M.R.A., Rizzo, G., Torchiano, M., Mihindukulasooriya, N., Corcho, O., García-Castro, R., 2019. Completeness and consistency analysis for evolving knowledge bases. Journal of Web Semantics 54, 48–71. URL: `https://www.sciencedirect.com/science/article/pii/S1570826818300623`, doi:`https://doi.org/10.1016/j.websem.2018.11.004`. managing the Evolution and Preservation of the Data Web.

[134] RDF_Data_Shapes_Working_Group, 2017. Shapes constraint language (shacl), in: W3C recommendation. URL: `https://www.w3.org/TR/2017/REC-shacl-20170720/`.

[135] Rettinger, A., Nickles, M., Tresp, V., 2009. Statistical relational learning with formal ontologies, in: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 286–301. doi:`https://doi.org/10.1007/978-3-642-04174-7_19`.

[136] Robaldo, L., 2021. Towards compliance checking in reified i/o logic via shacl, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, Association for Computing Machinery, New York, NY, USA. p. 215–219. URL: `https://doi.org/10.1145/3462757.3466065`, doi:`10.1145/3462757.3466065`.

[137] Robinson, I., Webber, J., Eifrem, E., 2015. Graph Databases: New Opportunities for Connected Data. 2nd ed., O'Reilly Media, Inc. doi:`10.5555/2846367`.

[138] Sagi, G., 2014. Formality in logic: From logical terms to semantic constraints. Logique et Analyse 57, 259–276. doi:`10.2143/LEA.227.0.3053506`.

[139] Saleh, D.R., Kartika, Y.A., Akbar, Z., Krisnadhi, A.A., Fatriasari, W., 2023. On generating shacl shapes from collective collection of plant trait data, in: Proceedings of the 2022 International Conference on Computer, Control, Informatics and Its Applications, Association for Computing Machinery, New York, NY, USA. p. 326–330. URL: `https://doi.org/10.1145/3575882.3575945`, doi:`10.1145/3575882.3575945`.

[140] Samuel, J., 2021. Shexstatements: Simplifying shape expressions for wikidata, in: Companion Proceedings of the Web Conference 2021, Association for Computing Machinery, New York, NY, USA. p. 610–615. URL: `https://doi.org/10.1145/3442442.3452349`, doi:`10.1145/3442442.3452349`.

[141] Santangelo, O.E., Gianfredi, V., Provenzano, S., 2022. Wikipedia searches and the epidemiology of infectious diseases: A systematic review. Data Knowledge Engineering 142, 102093. URL: `https://www.sciencedirect.com/science/article/pii/S0169023X22000842`, doi:`https://doi.org/10.1016/j.datak.2022.102093`.

[142] Savković, O., Kharlamov, E., Lamparter, S., 2019. Validation of shacl constraints over kgs with owl 2 ql ontologies via rewriting, in: Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A.J., Lopez, V., Haller, A., Hammar, K. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 314–329. doi:`https://doi.org/10.1007/978-3-030-21348-0_21`.

[143] Schaffenrath, R., Proksch, D., Kopp, M., Albasini, I., Panasiuk, O., Fensel, A., 2020. Benchmark for performance evaluation of shacl implementations in graph databases, in: Gutiérrez-Basulto, V., Kliegr, T., Soylu, A., Giese, M., Roman, D. (Eds.), Rules and Reasoning, Springer International Publishing, Cham. pp. 82–96. doi:`https://doi.org/10.1007/978-3-030-57977-7_6`.

[144] Schenner, G., Bischof, S., Comploi-Taupe, R., Steyskal, S., 2018. Integrating semantic web technologies and asp for product configuration.

[145] Schmidt, M., Hornung, T., Meier, M., Pinkel, C., Lausen, G., 2010. SP2Bench: A SPARQL Performance Benchmark. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 371–393. URL: `https://doi.org/10.1007/978-3-642-04329-1_16`, doi:`10.1007/978-3-642-04329-1_16`.

[146] Schmidt, M., Lausen, G., 2013. Pleasantly consuming linked data with rdf data descriptions. URL: `https://arxiv.org/abs/1307.3419`, doi:`https://doi.org/10.48550/arXiv.1307.341`, `arXiv:1307.3419`.

[147] Seifer, P., Lämmel, R., Staab, S., 2021. Progs: Property graph shapes language, in: Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P., Haller, A., Dragoni, M., Alani, H. (Eds.), The Semantic Web – ISWC 2021, Springer International Publishing, Cham. pp. 392–409. doi:`https://doi.org/10.1007/978-3-030-88361-4_23`.

[148] Senthilvel, M., Beetz, J., 2020. A visual programming approach for validating linked building data. doi:`10.14279/DEPOSITONCE-9977`.

[149] Shape_Expressions_Community_Group, 2019. Shape expressions language 2.1. URL: `https://shex.io/shex-semantics/index.html`.

[150] Shu, Y., 2018. A practical approach to modelling and validating integrity constraints in the semantic web. Knowledge-Based Systems 153, 29–39. URL: `https://www.sciencedirect.com/science/article/pii/S0950705118301862`, doi:`https://doi.org/10.`

1016/j.knosys.2018.04.021.

[151] Sirin, E., Tao, J., 2009. Towards integrity constraints in owl, in: Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529, CEUR-WS.org, Aachen, DEU. p. 79–88. doi:10.5555/2890046.2890055.

[152] Solbrig, H.R., Prud'hommeaux, E., Grieve, G., McKenzie, L., Mandel, J.C., Sharma, D.K., Jiang, G., 2017. Modeling and validating hl7 fhir profiles using semantic web shape expressions (shex). Journal of Biomedical Informatics 67, 90–100. URL: https://www.sciencedirect.com/science/article/pii/S1532046417300345, doi:https://doi.org/10.1016/j.jbi.2017.02.009.

[153] Spahiu, B., Maurino, A., Palmonari, M., 2018. Towards improving the quality of knowledge graphs with data-driven ontology patterns and shacl, in: WOP@ISWC. doi:10.3233/978-1-61499-894-5-103.

[154] Stavropoulos, T.G., Meditskos, G., Lazarou, I., Mpaltadoros, L., Papagiannopoulos, S., Tsolaki, M., Kompatsiaris, I., 2021. Detection of health-related events and behaviours from wearable sensor lifestyle data using symbolic intelligence: A proof-of-concept application in the care of multiple sclerosis. Sensors 21. URL: https://www.mdpi.com/1424-8220/21/18/6230, doi:10.3390/s21186230.

[155] Staworko, S., Boneva, I., Labra Gayo, J., Hym, S., Prud'hommeaux, E., Solbrig, H., 2015. Complexity and expressiveness of shex for rdf doi:10.4230/LIPIcs.ICDT.2015.195.

[156] Staworko, S., Wieczorek, P., 2019. Containment of shape expression schemas for rdf, in: Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Association for Computing Machinery, New York, NY, USA. p. 303–319. URL: https://doi.org/10.1145/3294052.3319687, doi:10.1145/3294052.3319687.

[157] Stolk, S., McGlinn, K., 2020. Validation of ifcowl datasets using shacl, in: LDAC. URL: https://api.semanticscholar.org/CorpusID:221461290.

[158] Studer, R., Grimm, S., Abecker, A., 2007. Semantic Web Services: Concepts, Technologies, and Applications. doi:10.1007/3-540-70894-4.

[159] Sy, M.F., Roman, B., Kerrien, S., Montero Mendez, D., Genet, H., Wajerowicz, W., Dupont, M., Lavriushev, I., Machon, J.A., Pirman, K., Neela Mana, D., Stafeeva, N., Kaufmann, A.K., Lu, H., Lurie, J., Fonta, P.A., Garcia Rojas Martinez, A., Ulbrich, A., Lindqvist, C., Jimenez, s., Rotenberg, D., Markram, H., Hill, S., 2023. Blue brain nexus: An open, secure, scalable system for knowledge graph management and data-driven science. URL: https://infoscience.epfl.ch/handle/20.500.14299/197266, doi:10.3233/SW-222974.

[160] Taghzouti, Y., Vachtsevanou, D., Mayer, S., Ciortea, A., 2022. A step toward semantic content negotiation.

[161] Tao, J., Sirin, E., Bao, J., McGuinness, D.L., 2010. Integrity constraints in owl, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press. p. 1443–1448. doi:10.5555/2898607.2898837.

[162] Thapa, R.B., Giese, M., 2021. A source-to-target constraint rewriting for direct mapping, in: Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P., Haller, A., Dragoni, M., Alani, H. (Eds.), The Semantic Web – ISWC 2021, Springer International Publishing, Cham. pp. 21–38. doi:https://doi.org/10.1007/978-3-030-88361-4_2.

[163] Thapa, R.B., Giese, M., 2022. Mapping relational database constraints to shacl, in: Sattler, U., Hogan, A., Keet, M., Presutti, V., Almeida, J.P.A., Takeda, H., Monnin, P., Pirrò, G., d'Amato, C. (Eds.), The Semantic Web – ISWC 2022, Springer International Publishing, Cham. pp. 214–230. doi:https://doi.org/10.1007/978-3-031-19433-7_13.

[164] Thornton, K., Solbrig, H., Stupp, G.S., Labra Gayo, J.E., Mietchen, D., Prud'hommeaux, E., Waagmeester, A., 2019. Using shape expressions (shex) to share rdf data models and to guide curation with rigorous validation, in: Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A.J., Lopez, V., Haller, A., Hammar, K. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 606–620. doi:https://doi.org/10.1007/978-3-030-21348-0_39.

[165] Thuluva, A.S., Anicic, D., Rudolph, S., 2018. Iot semantic interoperability with device description shapes, in: Gangemi, A., Gentile, A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan, J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International Publishing, Cham. pp. 409–422.

[166] Tomaszuk, D., 2017. Rdf validation: A brief survey, pp. 344–355. doi:10.1007/978-3-319-58274-0_28.

[167] Turki, H., Jemielniak, D., Taieb, M.A.H., Gayo, J.E.L., Aouicha, M.B., Banat, M., Shafee, T.M.A., Prud'hommeaux, E., Lubiana, T., Das, D., Mietchen, D., 2022. Using logical constraints to validate statistical information about disease outbreaks in collaborative knowledge graphs: the case of covid-19 epidemiology in wikidata. PeerJ Computer Science 8. URL: https://api.semanticscholar.org/CorpusID:252636723, doi:10.7717/peerj-cs.1085.

[168] Uschold, M., Grüninger, M., 1996. Ontologies: Principles, methods and applications. The Knowledge Engineering Review 11. doi:10.1017/S0269888900007797.

[169] Vidal, J.C., Rabelo, T., Lama, M., Amorim, R., 2018. Ontology-based approach for the validation and conformance testing of xapi events. Knowledge-Based Systems 155, 22–34. URL: https://www.sciencedirect.com/science/article/pii/S0950705118302065, doi:https://doi.org/10.1016/j.knosys.2018.04.035.

[170] Web_Ontology_Working_Group, 2004. Owl web ontology language. URL: https://www.w3.org/TR/owl-ref/.

[171] Werbrouck, J., Taelman, R., Verborgh, R., Pauwels, P., Beetz, J., Mannens, E., 2020. Pattern-based access control in a decentralised collaboration environment.

[172] Wright, J., Rodríguez Méndez, S.J., Haller, A., Taylor, K., Omran, P.G., 2020. Schímatos: A shacl-based web-form generator for knowledge graph editing, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 65–80. doi:https://doi.org/10.1007/978-3-030-62466-8_5.

[173] Xu, L., Yuan, Q., Wu, J., Liu, C., 2009. Ontology-based web service robustness test generation, in: 2009 11th IEEE International Symposium on Web Systems Evolution, pp. 59–68. doi:10.1109/WSE.2009.5632118.

[174] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S., 2015. Quality assessment for linked data: A survey. Semantic Web 7, 63–93. doi:10.3233/SW-150175.

[175] Zhao, X., Li, F., Yang, H., 2023. Rdfs(c) schema inconsistency checking based on a key instance and its query rewriting. IEEE Access 11, 6122–6132. doi:10.1109/ACCESS.2023.3234084.

[176] Zhu, R., Shimizu, C., Stephen, S., Zhou, L., Cai, L., Mai, G., Janowicz, K., Schildhauer, M., Hitzler, P., 2022. Sosa-shacl: Shapes constraint for the sensor, observation, sample, and actuator ontology, in: Proceedings of the 10th International Joint Conference on Knowledge Graphs, Association for Computing Machinery, New York, NY, USA. p. 99–107. URL: `https://doi.org/10.1145/3502223.3502235`, doi:`10.1145/3502223.3502235`.